

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Corso di Laurea Magistrale in Ingegneria Informatica

Protocolli per Gestione di Informazioni di Traffico per Reti Veicolari Ad Hoc

Tesi di Laurea in Sistemi Mobili M

Relatore:
Prof. Ing.
Paolo Bellavista

Tesi di Laurea di:
Federico Caselli

Correlatori:
Dott. Ing. Luca Foschini
Prof. Ing. Antonio Corradi

II Sessione
Anno Accademico 2013/2014

Indice

Introduzione	1
1 Gestione intelligente del traffico	3
2 Strategie di raccolta delle informazioni	12
2.1 Direzione	14
2.2 Protocollo decentralizzato	17
2.2.1 Data fusion	18
2.2.2 Formazione e ciclo di vita dei gruppi	19
2.3 Protocollo centralizzato	26
2.4 Classi di veicoli	29
3 Architettura del sistema	31
3.1 Ambiente di simulazione	31
3.1.1 Architettura complessiva di iTETRIS	32
3.1.2 SUMO	34
3.1.2 NS-3	35
3.1.4 iTETRIS Control System	37
3.1.5 Applicazioni utente iTetris	40
3.2 Architettura generale	43
3.2.1 Interfaccia con iCS	47
3.2.2 Modulo di campionamento	53
3.2.3 Controllore dei protocolli	56
3.2.4 Scambio di messaggi	59
4 Implementazione dei protocolli ITS	66
4.1 Protocollo decentralizzato	66
4.1.1 Protocolli di formazione dei gruppi	66
4.1.1.1 Strategia reattiva	69
4.1.1.2 Strategia proattiva	76
4.1.1.3 Mappatura dei nodi circostanti	80
4.1.2 Protocolli di gestione del ciclo di vita	84
4.1.2.1 Esplorazione dei nodi membri	86
4.1.2.2 Terminazione di un gruppo	90
4.1.2.3 Annessione ad un gruppo	92
4.1.2.4 Uscita da un gruppo	93
4.2 Protocollo centralizzato	95

4.2.1	Comunicazione fra i nodi	96
4.2.2	Raccolta dati	100
5	Risultati sperimentali	103
5.1	Scenari di simulazione utilizzati	103
5.2	Ottimizzazioni ad iTETRIS	105
5.3	Configurazione delle simulazioni	110
5.4	Risultati ottenuti dalle simulazioni	113
5.4.1	Indicatori di prestazione	114
5.4.2	Confronto fra le implementazioni	116
5.4.3	Confronto fra i protocolli	123
5.4.4	Analisi delle informazioni sul traffico raccolte	130
5.4.5	Differenti classi di veicoli	135
5.4.6	Basse penetrazioni	141
6	Analisi dei risultati e sviluppi futuri	152
6.1	Risultati ottenuti	153
6.2	Limiti del sistema	154
6.3	Sviluppi futuri	158
	Conclusioni	163
	Appendice A	165
	Bibliografia	171

Introduzione

Negli ultimi decenni la ricerca sugli Intelligent Transportation System (ITS) si è occupata di promuovere tecnologie finalizzate al miglioramento della sicurezza e della qualità della guida, all'incremento della funzionalità dei sistemi di trasporto ed alla riduzione delle emissioni di CO₂. Gli studi sulla realizzazione di un'infrastruttura stradale intelligente, cioè capace di raccogliere automaticamente informazioni sul traffico veicolare ed intervenire dinamicamente su di esso per realizzare una viabilità più efficiente e sicura, si sono da tempo concentrati sull'utilizzo di reti mobili veicolari (Vehicular Ad-hoc NETwork), in quanto la possibilità di mettere in comunicazione i mezzi di trasporto attraverso semplici interfacce wireless amplierebbe enormemente le potenzialità e l'efficienza di tali applicazioni.

Il presente lavoro di tesi si inserisce nel contesto appena esposto ed intende implementare un insieme di protocolli in ambito VANET, in grado di rilevare la presenza di veicoli in avvicinamento ad un impianto semaforico e di raccogliere quelle informazioni di stato che consentano all'infrastruttura stradale di ottenere una stima il più possibile veritiera delle attuali condizioni del traffico in ingresso per ciascuna delle direzioni monitorate in tale punto.

Il sistema che sarà sviluppato si occuperà di raccogliere informazioni sul traffico circoscritte ad un'area locale all'unità infrastrutturale che ne fa richiesta. Per ottenere una stima del flusso stradale in un particolare punto di interesse è infatti sufficiente monitorare un'area limitata, senza la necessità di realizzare complessi sistemi volti ad estenderne l'estensione. Questa caratteristica permette di semplificare i protocolli che saranno realizzati in quanto non avranno la necessità di implementare le complesse politiche di routing che caratterizzano le reti mobili veicolari nel caso sia richiesta la comunicazione fra veicoli lontani fra loro.

La tecnologia di trasmissione alla base delle reti veicolali, indispensabile per l'insieme di protocolli che si realizzeranno, ha ancora una diffusione molto limitata nel mondo reale. Occorrerà pertanto prestare particolare attenzione al funzionamento del sistema in condizioni di bassa percentuale di adozione, valutandone accuratamente

le performance in questi scenari. Saranno analizzate le informazioni raccolte per verificarne il comportamento in tali condizioni, cercando di individuare se è possibile ottenere una stima del flusso del traffico basata su dati indipendenti da queste variazioni.

COLOMBO è un progetto finanziato dalla commissione europea volto allo studio di metodi per migliorare di controllo del traffico stradale attraverso la ricerca di protocolli per monitorare il flusso di veicoli e per ottimizzare la gestione degli impianti semaforici sfruttando le opportunità offerte dall'utilizzo di reti veicoli. Particolare attenzione viene posta agli scenari caratterizzati da basse percentuali di adozione discussi, solitamente non considerati in altri studi su applicazioni ITS. Il presente lavoro di tesi si inserisce in questo progetto ed utilizzerà gli strumenti di simulazione e gli scenari urbani messi a disposizione.

L'architettura realizzata verrà testata all'interno di un ambiente urbano simulato nel quale la mobilità dei nodi di rete corrisponde a rilevazioni reali effettuate su alcune porzioni della città di Bologna. Le performance e le caratteristiche del sistema complessivo verranno analizzate e commentate sulla base dei diversi test condotti e si cercherà di evidenziarne i punti di forza o di debolezza a seconda dei diversi scenari e casi d'uso considerati, fornendo eventuali spunti utili per l'evoluzione del progetto sulla base dei risultati ottenuti e dell'esperienza maturata.

Si porrà particolare enfasi sulla necessità di produrre delle logiche in grado di svolgere un'euristica soddisfacente delle reali condizioni del traffico anche in presenza di una scarsa percentuale di penetrazione della tecnologia stessa, ipotizzando eventualmente il ricorso ad interfacce di comunicazione di classe inferiore o comunque eterogenee.

Il primo capitolo fornirà una veloce introduzione sulle possibilità offerte dai sistemi ITS in reti veicolari e sulle problematiche ad essi connesse. Il capitolo 2 definirà i concetti più importanti per il sistema che si vuole realizzare ed indicherà le principali strategie adottate per soddisfarne i requisiti. Nei capitoli 3 e 4 verrà discussa l'implementazione dell'architettura complessiva e dei diversi protocolli realizzati, mentre nei capitoli 5 e 6 verranno presentati e commentati i risultati ottenuti a seguito dei test condotti e saranno indicati eventuali sviluppi futuri.

Capitolo 1

Gestione intelligente del traffico

Una delle principali caratteristiche della moderna società è una progressiva e costante crescita della popolazione, sempre più concentrata in aree urbane estese e caratterizzate da una densità abitativa sempre più elevata. Nei paesi maggiormente sviluppati, questo processo di urbanizzazione ha portato nel tempo ad accrescere il numero di persone, merci e strumenti che necessitano quotidianamente di compiere numerosi spostamenti.

Come conseguenza di questa evoluzione, un numero sempre maggiore di città si trova a dover affrontare ogni giorno ingenti problemi legati all'incapacità di gestire una quantità ed un flusso di mezzi di trasporto in costante aumento. Queste problematiche si ripercuotono con un generalizzato innalzamento del tempo necessario ai diversi utenti stradali per raggiungere la propria destinazione, con un aumento del numero di incidenti automobilistici e con un crescente aggravarsi delle condizioni ambientali, fortemente minacciate dall'inquinamento acustico ed atmosferico. Occorre inoltre considerare l'impatto economico sempre più rilevante che deriva dal tempo perso a causa delle congestioni dagli utenti.

Tra le numerose contromisure adottate per arginare o comunque alleviare questo problema, sono emersi negli ultimi decenni numerosi studi e ricerche molto promettenti con il preciso obiettivo di incrementare l'efficienza delle infrastrutture stradali già esistenti attraverso un miglioramento dei sistemi automatici per il controllo del traffico. Tali sforzi si mostrano di particolare interesse in quanto consentono di ottenere sensibili progressi anche in assenza di costosi interventi di ristrutturazione e riprogettazione dei servizi di viabilità.

I sistemi maggiormente diffusi per il controllo automatico del traffico risultano essere gli impianti semaforici per la regolamentazione dell'accesso dei veicoli agli incroci stradali. Tradizionalmente, la modalità di funzionamento principalmente utilizzata per tali dispositivi prevede una temporizzazione fissa e prestabilita di una sequenza temporale di segnali semaforici per indicare ai veicoli che transitano in una direzione se è permesso loro l'accesso all'incrocio. Lo svantaggio principale di questo

comportamento è dato dal fatto che esso non tiene minimamente conto delle reali condizioni del traffico circostante, dando luogo molto spesso a situazioni di iniqua distribuzione degli accessi all'incrocio, con conseguente spreco di tempo e di carburante da parte degli utenti stradali, ma soprattutto con l'insorgere di congestioni e rallentamenti che provocano un deterioramento delle condizioni di viabilità ed un maggiore inquinamento ambientale.

Il modello di accesso a temporizzazione fissa è oggi considerato del tutto inadeguato e negli ultimi decenni sono stati sperimentati ed adottati sistemi intelligenti per il controllo del traffico [GSL07]. Questi prevedono di regolamentare il passaggio degli utenti in maniera dinamica, basandosi cioè sulle reali condizioni del traffico in una determinata area ed applicando sofisticati algoritmi di scheduling per ottimizzare al meglio il flusso di veicoli nelle diverse direzioni controllate, considerando anche parametri addizionali quali la fascia oraria, l'ordine di priorità di un particolare tratto stradale, la presenza di mezzi di trasporto pubblici, numero di corsie disponibili, etc.

Queste strategie evolute per il controllo degli accessi hanno tuttavia esigenza di poter conoscere in ogni momento una stima del flusso di vetture attualmente presente nei diversi tratti stradali che precedono il punto di incrocio nel quale l'impianto semaforico è collocato. Il sistema tradizionalmente più utilizzato per rilevare in maniera automatica la presenza di veicoli in punti strategici è il ricorso a sensori ad induzione (induction loop) installati pochi centimetri al di sotto del livello del manto stradale e capaci di indicare l'esistenza o meno di un veicolo sopra a essi. Altri dispositivi frequentemente utilizzati per tale scopo possono essere sensori a pressione, i quali offrono l'ulteriore vantaggio di fornire una stima della velocità corrente del veicolo rilevato.

I sistemi di monitoraggio del traffico mediante sensori interrati presentano purtroppo numerosi svantaggi che ne limitano l'efficacia e di conseguenza l'adozione: essi richiedono anzitutto costi di installazione e manutenzione relativamente elevati, trattandosi di dispositivi collocati al di sotto del livello stradale; inoltre, la loro affidabilità può essere compromessa da falsi positivi o falsi negativi. Un fattore che in ogni caso ne limita fortemente le possibilità di uso per impianti semaforici intelligenti è costituito dal fatto che il ricorso a queste tipologie di sensori non garantisce un input sufficiente per le logiche di controllo in quanto non sono in grado di fornire informazioni avanzate sullo stato del traffico riguardanti il numero di veicoli presenti nelle diverse carreggiate, la lunghezza di eventuali code, la velocità di spostamento, il tempo di attesa degli utenti, etc.

Per superare queste difficoltà, sono state proposte nel tempo diverse classi di sensori per la sorveglianza di un flusso di traffico piuttosto che di un semplice veicolo, alcuni esempi possono essere dati da telecamere collegate a complessi software di visione, sensori acustici oppure cellule ad infrarossi. La grande diffusione degli smartphone ha inoltre permesso la creazione di un nuovo metodo di monitoraggio del traffico che prevede l'utilizzo di un server centrale dedicato alla raccolta delle informazioni trasmesse in forma anonima da questi dispositivi utilizzando la rete cellulare. Sfortunatamente, tutti questi sistemi comportano una nutrita serie di svantaggi e limitazioni e la loro applicabilità risulta ad oggi ancora seriamente compromessa.

I sistemi di controllo degli accessi dedicati alla creazione ed all'applicazione delle politiche di gestione, sulla base delle informazioni raccolte da questi sensori stradali, sono caratterizzati da un'architettura di tipo centralizzato oppure decentralizzato.

Il modello centralizzato prevede l'utilizzo di un centro di controllo dedicato alla raccolta di tutte le informazioni ottenute dai sensori posizionati sulla rete stradale, con l'obiettivo di costruire una base di conoscenza completa sull'intera area monitorata. Le informazioni così raccolte sono poi utilizzate per la creazione e la valutazione di una politica di gestione delle luci semaforiche da utilizzare nelle diverse intersezioni controllate attraverso questo sistema. È anche prevista la possibilità di intervento manuale sulle decisioni prese dal sistema da parte degli addetti al controllo del traffico. Nonostante siano già utilizzati in molti ambienti urbani, questi sistemi sono decisamente costosi in quanto richiedono l'utilizzo di estese reti di comunicazione, la creazione di un centro di controllo che possa processare il grande numero di informazioni raccolte dai sensori e la necessità di personale qualificato dedicato al controllo.

La modalità decentralizzata di controllo delle politiche semaforiche non utilizza un centro di controllo centralizzato, ma si basa sullo scambio di informazioni fra dispositivi infrastrutturali vicini. Questa organizzazione comporta costi aggiuntivi derivanti dalla rete di comunicazione ed è di difficile espansione in quanto l'installazione di una nuova unità necessita dell'aggiornamento di tutte quelle vicine che devono avere conoscenza reciproca per realizzare un controllo ottimale del flusso del traffico.

Sistemi più complessi prevedono l'utilizzo di una architettura mista che utilizza un centro di controllo dedicato alla creazione di una politica generale che sarà adattata da ogni unità semaforica per ottimizzarla allo stato del traffico nel particolare incrocio stradale controllato.

Recentemente è stato introdotto l'uso del termine Intelligent Transportation System (ITS) per indicare quel vasto campo di ricerca che si occupa di promuovere tecnologie finalizzate al miglioramento della sicurezza e della qualità della guida, all'incremento della funzionalità dei sistemi di trasporto e dei servizi di viabilità, nonché all'ottimizzazione nell'uso dei carburanti per una migliore salvaguardia dell'ambiente. Gli sforzi di tali studi si spingono quindi ben oltre la realizzazione di soluzioni telematiche per il controllo del traffico, puntando più genericamente all'integrazione delle conoscenze e dei contributi raggiunti nel campo dell'elettronica, dell'ingegneria meccanica, delle telecomunicazioni e dell'informatica per la creazione e la manutenzione di un'infrastruttura stradale intelligente, cioè capace di promuovere la raccolta automatica di informazioni di stato sull'ambiente e lo scambio di comunicazioni tra veicoli, utenti e dispositivi fissi al fine di realizzare una gestione del traffico più efficiente e sicura.

Nell'ultimo decennio, le ricerche e le sperimentazioni di ITS più promettenti prevedono l'utilizzo di tecnologie di comunicazione wireless in grado di realizzare forme di comunicazione molto efficienti tra veicoli ed infrastruttura stradale (V2I), tra veicoli e altri veicoli vicini (V2V) oppure, più frequentemente, in entrambi i casi d'uso (V2X). Queste funzionalità richiedono sia l'installazione di appositi dispositivi di comunicazione nei punti strategici della rete stradale, sia la dotazione di apposite interfacce wireless all'interno dei singoli veicoli. Diversi enti internazionali e compagnie automobilistiche stanno investendo ingenti risorse nella sperimentazione di prototipi in grado di sfruttare queste opportunità. Grazie all'assegnamento di una banda dedicata da parte delle autorità europee nel 2008 ed alla recente standardizzazione di appositi protocolli di comunicazione ad opera dell'ETSI, a partire dal 2015 ci si aspetta una graduale immissione nel mercato consumer di autovetture dotate di sensori, attuatori e sistemi di comunicazione specificatamente pensati per applicazioni di ITS [Kra13].

L'installazione di interfacce di comunicazione wireless sui singoli veicoli permette la creazione di reti VANET (Vehicular Ad-hoc NETwork), ovvero reti caratterizzate da elevata mobilità nelle quali i nodi consistono di autovetture capaci di comunicare con altri loro pari o con eventuali dispositivi fissi (Road-side Unit, RSU) facenti parte dell'infrastruttura stradale. Le opportunità che questa tecnologia permette di sfruttare sono molteplici: i mezzi di trasporto più recenti dispongono di numerosi sensori in grado di raccogliere una vasta quantità di informazioni sullo stato del veicolo in termini di posizione, velocità, direzione e molto altro; con l'introdu-

zione delle reti VANET, la possibilità di scambiare, condividere, trasferire e processare questa base di conoscenza verso altre vetture o verso dispositivi infrastrutturali dedicati alla gestione del traffico amplierebbe enormemente la diffusione e l'efficacia delle applicazioni ITS precedentemente accennate.

La possibilità di mettere in comunicazione veicoli vicini e di permetterne il collegamento non solo al web ma anche all'infrastruttura che regola e gestisce il servizio di viabilità locale apre concretamente l'orizzonte a degli scenari di controllo del traffico intelligente precedentemente soltanto ipotizzabili: raccolta di statistiche sul traffico con conseguenti interventi automatici e decentralizzati nell'eventualità di congestioni, monitoraggio dei veicoli a scopi di sicurezza e coordinamento dei soccorsi in caso di emergenze, assistenza al guidatore attraverso avvisi o deviazioni suggerite, apparecchiature stradali capaci di reagire in tempo reale alle condizioni contingenti del flusso di vetture locale o addirittura sistemi di guida automatica sono solo alcune delle possibilità attualmente in corso di studio e standardizzazione.

Pur basandosi su un'idea concettualmente semplice, la realizzazione di sistemi ITS mediante reti VANET richiede uno sforzo considerevole in termini di ricerca e sperimentazione. Le caratteristiche intrinseche del canale fisico di comunicazione e l'elevata mobilità tra veicoli pongono diversi problemi ai protocolli di comunicazione richiesti, ulteriormente aggravati dalla necessità di coordinare e gestire reti fortemente dinamiche, mutevoli e completamente decentralizzate. Per qualsiasi architettura che si intende realizzare occorre oltretutto disporre di approfonditi studi e sperimentazioni in scenari reali o simulati in grado di poter quantificare il rapporto costi - benefici offerto dalla particolare soluzione proposta.

Per agevolare la ricerca nell'ambito delle reti VANET e semplificarne il processo di sviluppo, il progetto iTETRIS [Ith14] mette a disposizione una piattaforma aperta, flessibile e conforme agli standard europei ETSI pensata per facilitare la simulazione e la messa a punto di applicazioni ITS efficienti. Questo studio ha visto la collaborazione di diversi partner ed università a livello europeo e ha prodotto e distribuito una considerevole quantità di pubblicazioni e deliverable utili per la sperimentazione di scenari per il controllo del traffico intelligente. Questa piattaforma sarà ampiamente utilizzata durante questo lavoro di tesi per la creazione e l'implementazione delle soluzioni progettate e per la loro sperimentazione e valutazione.

COLOMBO [Coh14] è un progetto attualmente in corso e finanziato dall'unione europea che punta a recuperare ed ampliare gli studi ed i risultati ottenuti in iTETRIS per applicarli al problema del controllo del traffico alle intersezioni. Scopo di questo

progetto è quello di ricercare e sperimentare un insieme di algoritmi e soluzioni per la sorveglianza del traffico e per la gestione dei flussi di veicoli in corrispondenza di incroci regolati da impianto semaforico. A differenza della maggior parte delle applicazioni studiate che sfruttano la comunicazione fra veicoli per modificarne il comportamento in base alle informazioni sul traffico raccolte ed utilizzano le unità infrastrutturali solamente come supporto per la trasmissione dei messaggi, l'idea alla base di COLOMBO è quella di utilizzare questo scambio di informazioni fra nodi mobili e fissi per modificare il comportamento delle unità semaforiche sulla base dello stato del traffico rilevato dai veicoli.

Simili ricerche condotte in passato hanno proposto alcuni approcci a questo problema e questi sono risultati in genere piuttosto efficienti purché si ammetta di disporre di una alta penetrazione di utenti dotati di veicoli equipaggiati con i sistemi e le architetture studiate. Una caratteristica originale di questo progetto è la particolare attenzione prestata alla realizzazione di soluzioni specificamente progettate per poter funzionare anche in presenza di basse percentuali di penetrazione della tecnologia utilizzata. Queste soluzioni prevedono l'utilizzo di dispositivi eterogenei per la raccolta di informazioni composti sia da apparati realizzati ad hoc installati all'interno delle autovetture sia da smartdevice personali degli utenti. Questi dispositivi portatili, grazie alla grande evoluzione che li ha caratterizzati negli ultimi anni, offrono i sensori di posizionamento e le interfacce di comunicazione necessari per partecipare nello scambio di dati realizzato.

Il lavoro di tesi presentato in questo testo si inserisce nel contesto di questo progetto europeo e parte dall'esigenza di riuscire a realizzare un insieme di protocolli in grado di rilevare la presenza di autovetture in avvicinamento ad un generico semaforo e di raccogliere delle informazioni di stato che consentano a tale RSU (cioè al dispositivo semaforico) di ottenere una stima il più possibile veritiera delle attuali condizioni del flusso di veicoli provenienti dai diversi sensi di marcia previsti per tale incrocio.

I dati prodotti dal sistema che verrà di seguito proposto ed analizzato costituiranno l'input di una successiva logica collocata internamente al semaforo ed incaricata dell'applicazione delle apposite politiche intelligenti di monitoraggio e di gestione dinamica del traffico. Lo studio qui realizzato si concentrerà sulle modalità con cui è possibile recuperare ed integrare tali informazioni, tralasciando i dettagli riguardanti le logiche che si occuperanno in un secondo momento di utilizzarle per intervenire in tempo reale sulle condizioni della rete stradale.

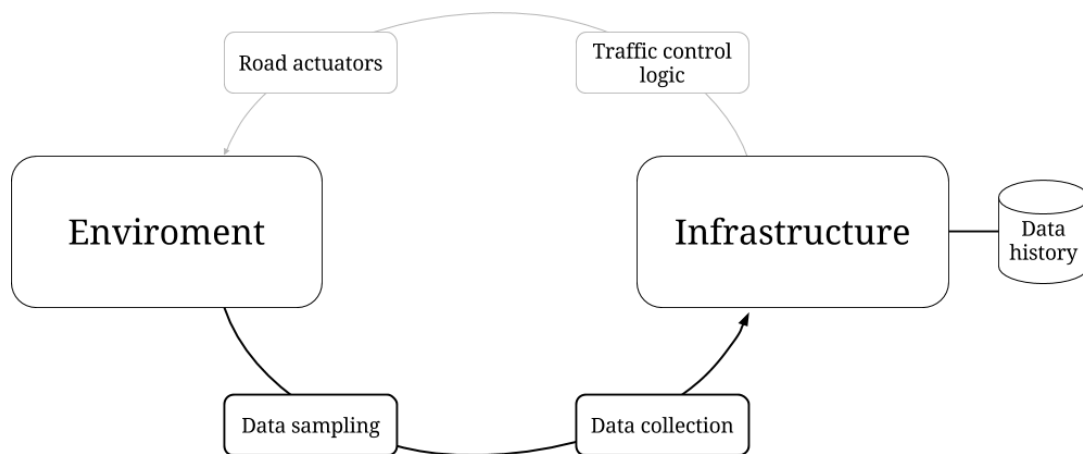


Figura 1.1: Schematizzazione ad alto livello del ciclo di gestione del traffico veicolare mediante un generico sistema ITS. I blocchi concettuali sui quali si è concentrato il presente lavoro di tesi vengono evidenziati in grassetto.

Il dominio applicativo del presente lavoro di tesi viene schematizzato sinteticamente in Figura 1.1 dove viene illustrato il processo ad alto livello secondo il quale un sistema ITS interagisce con l’ambiente circostante per realizzare le proprie funzionalità. Il contesto applicativo di riferimento è una predefinita area urbana contenente l’incrocio che si intende monitorare e dalla quale si estraggono le informazioni relative alle condizioni del traffico di veicoli. Questi dati vengono opportunamente consegnati ai dispositivi dell’infrastruttura stradale che si preoccupano di intervenire in tempo reale sul sistema stesso nei modi e nei tempi previsti dalle particolari politiche di gestione osservate, eventualmente sfruttando uno storico delle rilevazioni passate. Attraverso continui processi di azione e retroazione, il sistema si sforza nel tempo di mantenere le condizioni del traffico veicolare in costante equilibrio.

Un sistema con un funzionamento conforme a quanto appena descritto può ovviamente ricorrere a svariate tecnologie per l’approvvigionamento dei dati in ingresso; quelli recuperabili sfruttando strategie di tipo VANET risultano tuttavia sensibilmente più completi ed accurati rispetto a quelli ottenibili servendosi dei tradizionali sensori ad induzione o a pressione oppure ricavabili attraverso complessi sistemi di visione. Infine, la dotazione di interfacce wireless e tecnologie ITS sui singoli veicoli, pur richiedendo ingenti sforzi e risorse da investire in ricerca, sperimentazione e standardizzazione, risulta a lungo termine molto più conveniente in termini di rapporto vantaggi - costi rispetto all’installazione dei tradizionali dispositivi di rilevazione precedentemente accennati, i quali richiedono continui e costosi interventi di manutenzione stradale uniti ad una costante supervisione manuale da personale specializzato.

Saranno realizzate due strategie di raccolta dei dati dall'ambiente. Una prevedrà di organizzare i veicoli in gruppi ogni volta questi si avvicinano a un incrocio controllato da una RSU. Ogni gruppo sarà costituito esclusivamente da quelle vetture che transitano sullo stesso tratto stradale e promuoverà l'interazione tra i suoi diversi membri per raccogliere ed integrare i dati dei veicoli che vi partecipano. Queste informazioni verranno successivamente inviate alla RSU che provvederà ad rielaborare e registrare le statistiche in ingresso, organizzandole in base alla specifica direzione di provenienza. Per poter raggruppare i veicoli e dirigere le loro interazioni è necessario predisporre degli appositi protocolli per la formazione di un gruppo, oltre a quelli indispensabili per la gestione del suo ciclo vitale e per lo svolgimento dei compiti assegnati.

La seconda strategia prevede una comunicazione diretta fra i singoli nodi ed i dispositivi infrastrutturali, nella quale viene utilizzata solamente la tipologia di comunicazione V2I in quanto i veicoli non sono consapevoli dei loro pari ed agiscono in maniera indipendente gli uni dagli altri. Le unità semaforiche ricevono dai nodi mobili un flusso di informazioni frammentato ma molto completo, e si occuperanno di rielaborare queste informazioni per ottenere statistiche ed indicazioni utili ad attuare la politica semaforica desiderata. Per evitare una comunicazione caotica e disorganizzata delle informazioni i dispositivi infrastrutturali saranno incaricati in prima persona del coordinamento dei nodi.

Nel capitolo seguente verranno meglio definiti i concetti di rilevanza fondamentale per il sistema e si indicheranno le principali scelte adottate per soddisfare i requisiti appena esposti.

Queste due strategie saranno implementate all'interno della piattaforma iTE-TRIS attraverso lo sviluppo di un componente che si occuperà di gestire tutte le interazioni fra il simulatore e le logiche studiate. Questo modulo ha lo scopo di facilitare l'implementazione dei protocolli di comunicazione studiati in questo lavoro di testi, e fornisce una base che permette l'estensione dei protocolli realizzati e la creazione di nuovi.

Le varie logiche di protocollo implementate nel corso di questo progetto verranno presentate nel capitolo 4 e costituiscono il fulcro principale del sistema complessivo. Nel capitolo 3 saranno invece esposti i componenti che costituiscono la piattaforma di base per l'architettura prodotta e gli strumenti software utilizzati per implementarli e per simulare un ambiente di rete su scala urbana. Infine, nei capitoli 5 e

6 saranno discussi e commentati i risultati ottenuti a seguito dei test condotti, concludendo con un'analisi dei punti di forza e delle imperfezioni e suggerendo eventuali sviluppi futuri volti al perfezionamento ed all'ampliamento del lavoro realizzato.

Capitolo 2

Strategie di raccolta delle informazioni

Scopo di questo capitolo è quello di presentare le soluzioni risolutive che sono state considerate e le scelte affrontate per soddisfare i requisiti presentati nelle precedenti pagine. Si analizzerà il problema ad alto livello, trascurando in questa fase i dettagli non strettamente correlati all'individuazione di una corretta strategia implementativa. Per un'analisi approfondita delle problematiche legate alla realizzazione concreta dei protocolli proposti, delle scelte architettureali e delle modalità di simulazione e sperimentazione si rimanda ai capitoli successivi.

Volendo riassumere brevemente le funzionalità che il progetto si propone di soddisfare si ricorda come la principale esigenza che ha ispirato l'intero lavoro di tesi sia quella di poter fornire a dei dispositivi posti in corrispondenza a dei punti di viabilità critici (principalmente impianti semaforici posti a regolazione di incroci stradali) informazioni relative ad una stima del traffico dei veicoli in arrivo, opportunamente differenziate in base allo specifico senso di provenienza. La raccolta di simili informazioni può essere in generale utilizzata per ricavare nel lungo periodo delle statistiche sul flusso di veicoli che attraversa una porzione di rete stradale di particolare interesse. Più concretamente questi rilevamenti possono essere sfruttati per informare in tempo reale ogni dispositivo semaforico (RSU) della quantità di veicoli che sopraggiungono da ciascuna direzione di traffico, in modo da poter realizzare delle politiche di regolamentazione del passaggio più intelligenti e proattive rispetto alla semplice temporizzazione dei segnali semaforici.

I requisiti minimi per poter operare in questo scenario prevedono che i dispositivi dispongano di una strumentazione GPS o equivalente per la determinazione più o meno precisa della propria posizione e di almeno un'interfaccia di comunicazione remota per lo scambio di messaggi. La maggior parte delle ricerche riguardanti network veicolari si concentrano sull'utilizzo di 802.11 come principale standard di rete. In particolare, 802.11p e 802.11s risultano essere gli stack protocollari specificatamente pensati per un utilizzo nel campo delle reti VANET, in particolare per le loro caratteristiche di QoS e per la possibilità di realizzare reti mesh e protocolli di geo-routing

tra nodi ad elevata mobilità. Oltre alle soluzioni sopra citate, il recente successo ed enorme diffusione di dispositivi portati come PDA, smartphone, tablet ed i primi esempi di accessori wearable hanno spinto notevolmente l'evoluzione dei tradizionali apparati di rete 802.11b/g. Le crescenti necessità di coprire con la stessa tecnologia WiFi spazi di lavoro, ambienti domestici e scenari di tipo urbano, hanno portato alla ratificazione degli standard 802.11n e 802.11ac [PG11] in grado rispettivamente di raddoppiare e triplicare il raggio di copertura delle soluzioni wireless tradizionali di tipo 802.11g, oltre a garantire un flusso di dati più veloce, latenze minori ed un maggior numero di funzionalità. A fronte di simili progressi, viene naturale ipotizzare che tali interfacce possano vedere in futuro un'ampia adozione anche a bordo dei tradizionali autoveicoli consumer destinati al mercato di massa. A supporto di questa previsione è possibile notare come alcune case automobilistiche stiano già investendo nella ricerca sull'integrazione di diverse tecnologie wireless (Bluetooth, UMTS e più recentemente LTE e WiFi) a bordo di autovetture di ultima generazione, anche se per il momento la motivazione principale è quella di soddisfare esigenze di comunicazione, infotainment e sicurezza stradale [Jur12; CSZ11].

È comunque necessario specificare che le soluzioni proposte in questo testo non presuppongono l'utilizzo di un particolare standard 802.11 rispetto ad altri: la motivazione è dovuta principalmente al fatto che le comunicazioni tra i diversi nodi, come verrà meglio precisato più avanti in questo capitolo, avvengono esclusivamente mediante l'invio di semplici messaggi di tipo broadcast in configurazioni wireless ad-hoc. Non vengono pertanto sfruttate quelle funzionalità di routing, QoS o controllo degli accessi che costituiscono le caratteristiche che contraddistinguono gli standard 802.11 più avanzati.

È importante sottolineare a questo punto come le informazioni ricavate per mezzo degli strumenti presentati costituiranno soltanto una stima imperfetta ed approssimata del reale flusso di veicoli in transito verso la RSU: esistono infatti una moltitudine di variabili, fattori avversi ed eventi imprevedibili che possono compromettere la qualità dell'euristica fornita quali l'eterogeneità dei dispositivi in uso ed il loro livello di penetrazione, imperfezioni nei protocolli realizzati, errori ed interferenze nella trasmissione dei messaggi, comportamenti inattesi degli utenti del servizio, condizioni climatiche sfavorevoli che riducono la portata e la qualità delle trasmissioni, generici malfunzionamenti e molto altro ancora.

Alla luce di questi presupposti, qualsiasi soluzione proposta in ambito VANET deve tenere conto, oltre che dell'elevata mobilità ed eterogeneità dei dispositivi coin-

volti, delle condizioni tipicamente avverse in cui ci si trova a dover operare. I protocolli di seguito sviluppati cercheranno dunque di tollerare il più possibile eventuali perdite di pacchetto. Qualora risulti necessaria una maggiore garanzia che un destinatario riceva delle informazioni di particolare criticità verrà richiesta la conferma dei dati ricevuti mediante appositi messaggi di acknowledgement.

2.1 Direzione

In precedenza è già stato anticipato che la soluzione in esame presuppone la capacità da parte di ogni nodo di ottenere adeguate informazioni sulla propria posizione e sulla velocità corrente, sia pure con un certo grado di incertezza e imprecisione. In assenza di sensori specifici in grado di rilevare la velocità istantanea del veicolo è comunque possibile ricavare questa grandezza esaminando una serie di campionamenti della posizione effettuati a distanza sufficientemente ravvicinata e calcolando una media del rapporto tra la distanza che intercorre fra due campioni ed il tempo che li separa. Un altro importante parametro che risulta facilmente estraibile dal campionamento nel tempo della propria posizione è la direzione corrente. Questo concetto ricopre un ruolo cruciale nell'impostazione complessiva dell'intero sistema, è quindi di fondamentale importanza definirlo con chiarezza e capire come e perché questa informazione verrà utilizzata per differenziare i vari veicoli presenti nell'ambiente.

Dovendo mettere a confronto le direzioni dei diversi veicoli è indispensabile disporre di un sistema di riferimento appropriato per poter operare con questa grandezza. In generale il problema non è di facile soluzione: nonostante sia possibile con buona approssimazione assegnare ad un oggetto posto sulla superficie terrestre una posizione univoca, non è tuttavia immediato individuare un metodo che consenta di confrontare in ogni momento l'orientamento di due o più mezzi di trasporto. Per evitare di ricorrere a complessi sistemi geodetici è sufficiente osservare che nei casi d'uso previsti per le normali applicazioni di gestione del traffico, ed in particolare per quella in esame, non vengono mai messe in atto delle interazioni tra nodi talmente distanti da dovere essere considerati come vetture in movimento su una superficie curva. Anche se si prevedesse l'utilizzo delle soluzioni qui esposte per finalità ulteriori rispetto allo studio del flusso di autovetture, sarebbe perfettamente accettabile posizionare queste ultime in un piano bidimensionale con l'asse delle ascisse positive solidale all'est geografico. Un simile riferimento è del tutto analogo al sistema locale East,

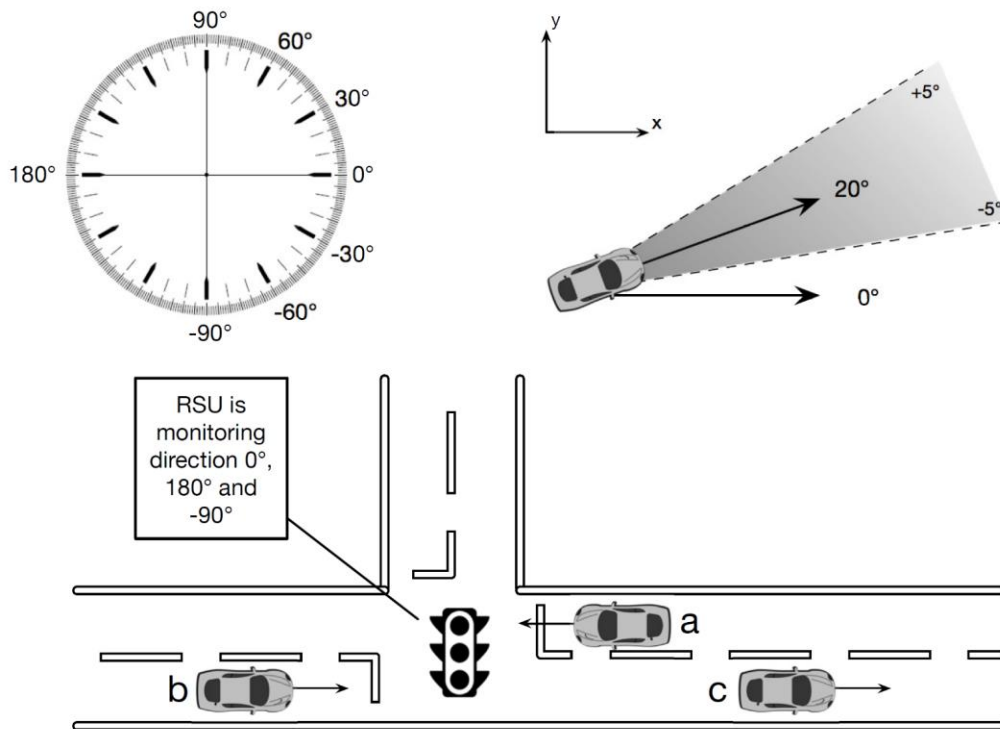


Figura 2.1: Definizione di direzione di un veicolo, cono di conformità ed applicazione in uno scenario esemplificativo.

North, Up (ENU¹) definito in [Dan03] e [Tor01]. Il passaggio dalle coordinate di tipo geografico utilizzate nei comuni dispositivi GPS alle coordinate cartesiane in accordo con il riferimento appena esposto risulta una procedura piuttosto semplice. Se si considerano delle grandezze che stimano una distanza tra due punti questo accorgimento è ovviamente trascurabile, quando viceversa occorre lavorare con una posizione assoluta è possibile definire l'origine in base alle proprie esigenze.

Avendo definito un sistema di riferimento cartesiano per il posizionamento dei veicoli, il problema di assegnare una direzione univoca a qualsiasi oggetto in movimento è di facile soluzione e viene affrontato come descritto in Figura 2.1: la direzione viene espressa in gradi d'arco compresi tra -180° e 180° . La direzione 0° è definita come quella dell'asse delle ascisse crescenti (est geografico) e l'angolo che il vettore velocità di un qualsiasi veicolo forma con esso determina il valore di tale grandezza; le direzioni 180° e -180° vengono considerate equivalenti. Ovviamente ad un mezzo di trasporto fermo o con velocità talmente bassa da non risultare percepibile dagli

¹ Per maggiori informazioni http://en.wikipedia.org/wiki/Geodetic_system

strumenti non ha senso assegnare un orientamento, tuttavia è possibile rimediare a questo problema conservando l'ultima grandezza valida registrata.

Si consideri l'esempio in Figura 2.1 dove una RSU (semaforo) controlla l'accesso dei veicoli alle tre strade raffigurate: è ragionevole supporre che i dispositivi di infrastruttura conoscano a priori un vettore di valori corrispondenti alle direzioni da cui si desidera ricevere degli aggiornamenti di stato, visto che questi dipendono dalla particolare topologia stradale in quel punto e rimangono fissi nel tempo. Nel nostro caso questi parametri dovrebbero valere 180° per i veicoli provenienti da destra (il nodo a nell'esempio), 0° per quelli che raggiungono l'incrocio da sinistra (nodo b) e -90° per quelli che provengono da nord. In generale questo sistema può essere applicato a qualsiasi tipo di incrocio o snodo stradale comunemente utilizzato, si noti comunque che occorre assicurarsi che i segni degli angoli siano scelti in accordo alle direzioni previste per i veicoli che si dirigono verso una RSU. Terminando l'analisi dell'esempio fornito, si osservi come il veicolo c non può partecipare a un ipotetico plotone che comprende il nodo a in quanto le direzioni dei due veicoli non corrispondono; esso non deve inoltre consegnare alcun dato alla RSU visto che ha già attraversato l'incrocio e si sta allontanando, tuttavia la sua direzione è compatibile con quelle monitorate. Questa complicazione è risolvibile tenendo conto che il veicolo in questione deve avere necessariamente attraversato l'incrocio in precedenza e ha già svolto un'interazione con il dispositivo, ulteriori comunicazioni possono essere scongiurate prevedendo una lista di RSU già visitate ed escludendo temporaneamente queste ultime dai protocolli utilizzati dai nodi.

Esiste tuttavia un ultimo problema che è necessario affrontare: in uno scenario realistico, durante la marcia, un veicolo effettua continui ed imprevedibili cambi di direzione dovuti a molteplici concause: imperfezioni nella convergenza dei treni di ruote, leggere correzioni allo sterzo da parte del guidatore, tentativi di sorpasso, dissesto stradale e molte altre. Un'autovettura che si spostasse da ovest verso est seguendo una strada perfettamente parallela all'equatore terrestre difficilmente riuscirebbe a mantenere una direzione esattamente pari a 0° , ma misurerebbe nel tempo dei valori che si discostano leggermente da esso. Oltre a questi fattori si deve anche tenere conto di inevitabili errori introdotti dai sensori utilizzati per monitorare lo stato del veicolo. È opportuno quindi prevedere uno o più meccanismi per compensare queste variazioni indesiderate ed evitare dei falsi negativi nel momento in cui la direzione di un veicolo viene confrontata con un angolo di riferimento o con la traiettoria di un altro nodo. Una prima semplice ma efficace accortezza è quella di mantenere per un

opportuno periodo di tempo una storia dei precedenti campionamenti relativi alla direzione del veicolo e di combinarli effettuandone una media circolare², in questo modo è possibile contenere buona parte dei disturbi introdotti dalle imprecisioni degli strumenti di misura o da leggeri cambiamenti di traiettoria.

Una soluzione aggiuntiva al problema viene presentata in Figura 2.1: ogniqualvolta sia necessario confrontare tra loro due misure di orientamento (ad esempio per verificare se due veicoli stanno procedendo nella stessa direzione) queste si possono considerare coincidenti anche a fronte di una lieve differenza, purché il loro disallineamento non superi una soglia stabilita a priori. Se il veicolo in alto a destra nella figura viaggia con una direzione media di 20° , questo potrà interagire anche con altri nodi che procedono con un orientamento che non si discosta di oltre 5° in valore assoluto dal suo. Una coppia di valori di direzione che mantengono questa proprietà viene definita conformante. Nella soluzione sviluppata questo importante accorgimento è stato utilizzato in diverse occasioni, in particolare per consentire ai diversi protocolli di continuare a funzionare anche in caso di percorsi leggermente irregolari.

Il concetto di direzione conforme è particolarmente utile non soltanto come semplice strumento per individuare e selezionare quei veicoli che rispettano un particolare senso di marcia ma anche come tecnica con la quale è possibile realizzare una comunicazione di tipo multicast tra i diversi nodi presenti nell'ambiente. Un qualsiasi dispositivo o interfaccia che intende scambiare informazioni con un insieme di nodi non necessariamente noti a priori, potrebbe imporre che la direzione attuale dei destinatari debba sottostare a precise condizioni, quali ad esempio la conformità ad un dato senso di marcia in modo da circoscrivere lo scambio di informazioni tra i soli veicoli che condividono la stessa direzione. Questa tecnica di selezione dei destinatari di un messaggio sarà applicata sistematicamente nell'implementazione dei diversi protocolli realizzati.

2.2 Protocollo decentralizzato

Il protocollo descritto in seguito è stato originariamente sviluppato da Enrico Zamagni. Il lavoro svolto per questa tesi è stato finalizzato alla realizzazione di una implementazione all'interno della piattaforma iTETRIS. Nei capitoli seguenti saranno illustrate le caratteristiche dei simulatori utilizzati e verrà approfondito il lavoro di implementazione.

² Per maggiori dettagli sul concetto di media circolare http://en.wikipedia.org/wiki/Mean_of_circular_quantities

2.2.1 Data fusion

Con il termine data fusion si intende descrivere il processo di integrazione di un vasto gruppo di dati di basso livello in un insieme di informazioni più generali e rappresentabili in maniera più leggibile, in modo da poter usufruire di una visione più ampia ed utile del particolare contesto che si vuole esaminare [HL01]. Il termine è in realtà piuttosto generico e viene spesso utilizzato impropriamente per indicare una qualsiasi serie di trasformazioni operate su una vasta collezione di dati con lo scopo di ottenerne una visione di più alto livello o comunque maggiormente riassuntiva o human-readable. In questo testo intenderemo con data fusion la raccolta di un grande quantitativo di dati ad opera dei singoli veicoli e la loro successiva integrazione in una struttura informativa che ne riassume in maniera più sintetica le proprietà di maggiore interesse. Anticipando i concetti che verranno di seguito esposti, possiamo prevedere che le informazioni di più basso livello riguardino lo stato del singolo veicolo facente parte del sistema e possono comprendere i dati ottenuti mediante i diversi dispositivi di cui esso è dotato: la sua posizione, la sua velocità o la presenza di eventuali altri nodi vicini; questa vasta base di conoscenza viene quindi raccolta ed elaborata per ricavare un'euristica sull'attuale condizione del traffico e sul comportamento degli utenti che lo compongono [FLK11].

I numerosi veicoli sparsi nello scenario di Figura 2.2 sono in grado individualmente di raccogliere una vasta quantità di informazioni su se stessi e sull'ambiente che li circonda. Questo insieme di dati può risultare però troppo ampio e dettagliato per potere essere direttamente utilizzato come stima delle condizioni del flusso di veicoli in una determinata zona o verso una specifica direzione. Inoltre, il trasferimento di un quantitativo di informazioni così abbondante verso un unico dispositivo si potrebbe rivelare difficoltoso in condizioni di elevata densità ed occorrerebbe garantire la capacità per poter processare in tempo reale una base di conoscenza di dimensioni considerevoli.

Come si nota osservando la Figura 2.2, questa funzionalità viene realizzata introducendo un livello di indirettezza aggiuntivo tra i dati generati dai singoli veicoli e la loro assimilazione in forma più compatta, contestualizzata ed astratta da parte dei livelli applicativi più elevati. Una RSU (stilizzata con l'icona di un semaforo) non viene più informata dello stato di ogni singolo veicolo in avvicinamento, ma riceve piuttosto informazioni sulla presenza di uno o più insiemi di veicoli e sulle loro proprietà. Questa maggiore granularità dei dati, integrati in maniera distribuita dai vari

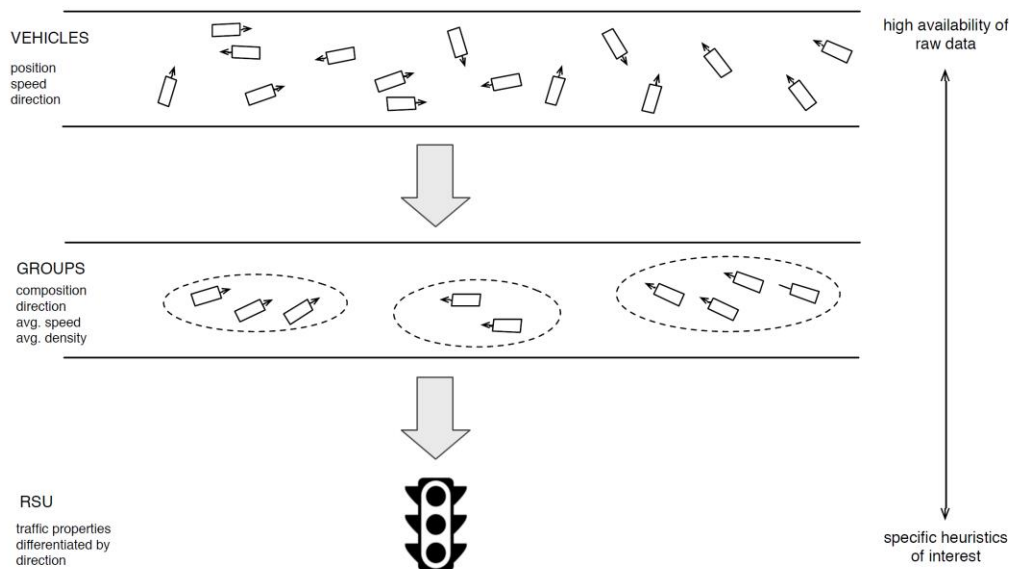


Figura 2.2: Raggruppamento dei veicoli in gruppi per fornire alla RSU informazioni aggregate di più alto livello.

agenti del sistema e non dalla singola RSU, consente di spostare buona parte del carico computazionale dalle unità infrastrutturali fisse verso i dispositivi mobili.

Come abbiamo già avuto occasione di specificare, l'impianto architetturale presentato in questo testo non ha unicamente lo scopo di rispondere alle problematiche di gestione del traffico già esposte in precedenza, ma vuole anche essere un punto di partenza flessibile per soluzioni ITS di diverse tipologie. Questa visione granulare di alto livello dei dati raccolti può però portare ad una limitazione della flessibilità di questa architettura a causa della difficoltà di espansione e modifica della logica che si trova distribuita nei vari livelli dell'architettura. Inoltre, considerando la scarsa diffusione delle tecnologie V2X, che riduce le potenzialità offerte dal raggruppamento dei veicoli a causa dell'elevato numero di gruppi composti da un singolo dispositivo, e la mancanza di standard per soluzioni ITS, che può portare a frequenti modifiche alle capacità richieste all'architettura, ha portato a considerare anche il protocollo centralizzato descritto nel paragrafo 2.3.

2.2.2 Formazione e ciclo di vita dei gruppi

Nel precedente paragrafo è stato illustrato il principio cardine sul quale si fonda il protocollo decentralizzato: la volontà di integrare i dati di basso livello allo scopo

di fornire alle RSU delle informazioni relative ad uno o più insiemi di veicoli. È necessario a questo punto chiarire su quali basi si intende assegnare ciascun nodo a un preciso gruppo piuttosto che ad un altro. L'esigenza è che i dati aggregati di ogni gruppo contribuiscano a delineare lo stato complessivo del flusso del traffico in un determinato senso di marcia: per poter applicare delle politiche intelligenti di regolazione del traffico stradale non è sufficiente che il dispositivo che regola l'accesso dei veicoli a uno snodo conosca il numero di vetture in avvicinamento ma è cruciale che tale informazione, unita ad altre, sia contestualizzata in base alla direzione che queste stanno seguendo.

Basandosi su tutti i requisiti esposti fino a questo momento, si può definire il concetto di gruppo sfruttando i due concetti di distanza e direzione: stabiliamo quindi che un insieme di veicoli costituisce un gruppo soltanto se questi sono accomunati dal fatto di essere fisicamente vicini gli uni agli altri e di possedere lo stesso senso di marcia. Questa strategia di raggruppamento non costituisce una novità essendo del tutto analoga all'idea di plotone frequentemente esposta in letteratura ed utilizzata in numerosissime sperimentazioni in ambito VANET. Nel nostro contesto, i due termini sono intercambiabili e verranno utilizzati senza alcuna distinzione.

Una soluzione che si ponesse l'obiettivo di essere versatile e completamente personalizzabile potrebbe semplicemente assumere che due veicoli risultino vicini quando la loro distanza reciproca non supera una certa soglia stabilita in fase di configurazione del sistema. Questo permetterebbe di adattarsi a quei casi d'uso dove dei gruppi di dimensione troppo estesa potrebbero risultare indesiderati o problematici da gestire e si preferisce avere il massimo controllo sui criteri utilizzati per la creazione dei plotoni. Questa funzionalità potrebbe però portare a molteplici difficoltà di implementazione dal momento che occorre tenere conto del fatto che la capacità di comunicazione di ogni nodo, inclusa la stessa RSU, è di fatto limitata dalla portata del segnale utilizzato per la trasmissione dei dati, una grandezza che può variare in maniera imprevedibile da dispositivo a dispositivo e che si trova ad essere fortemente influenzata dalle contingenti condizioni dell'ambiente. Nel momento in cui venisse richiesta un'estensione dei plotoni superiore al raggio di copertura dei veicoli che lo compongono, si dovrebbe necessariamente fare a ricorso ad un meccanismo di scambio dei dati che consenta ad ogni veicolo di consegnare un messaggio destinato ad un qualsiasi altro membro di quello stesso gruppo, indipendentemente dalla distanza che intercorre tra essi e servendosi eventualmente di uno o più nodi intermediari per recapitare il contenuto nel caso in cui un'eccessiva separazione tra mittente e destinatario

non consenta uno scambio di informazioni diretto. Questo problema è stato ampiamente studiato ed esiste una letteratura [HXG02] molto vasta ed approfondita sui diversi protocolli ed algoritmi di routing che possono essere applicati al fine di permettere la comunicazione tra nodi distanti, soprattutto nel campo delle reti veicolari [ASP08]. In ogni caso, a causa dell'elevata mobilità tra i nodi in esame, quasi tutte le soluzioni disponibili sono di tipo best-effort e nessuna di esse è in grado di garantire in ogni momento uno scambio di dati efficace e stabile. Inoltre, per via dei possibili pattern di mobilità differenti con cui si spostano i loro membri, i plotoni possono sempre andare incontro a partizionamenti e perdite di connettività che possono seriamente compromettere le loro interconnessioni.

Avendo precedentemente dichiarato l'intenzione di sviluppare un'architettura il più possibile semplice e poco costosa, l'introduzione di avanzati protocolli di routing tra i partecipanti di un gruppo risulterebbe eccessiva e porterebbe a dover gestire una complessità tale da non essere giustificata a fronte di funzionalità che raramente risulterebbero utili per i casi d'uso previsti: per arrivare ad una buona stima del flusso del traffico che anticipa uno snodo stradale non occorre infatti prevedere la formazione di plotoni di veicoli particolarmente estesi ed in condizioni di elevata densità o congestione, gruppi dal diametro relativamente contenuto riusciranno comunque a suggerire alla RSU la criticità dello stato attuale.

Una soluzione diametralmente opposta potrebbe imporre che una qualsiasi coppia di nodi deve essere in comunicazione diretta per poter appartenere allo stesso gruppo. Questo risolverebbe alla radice qualsiasi problema di comunicazione tra veicoli distanti ma risulta essere una condizione eccessivamente vincolante visto che sotto simili presupposti l'estensione massima di un plotone sarebbe sempre limitata al più piccolo tra i raggi di portata dei singoli membri. I gruppi formati seguendo tale direttiva potrebbero rivelarsi di dimensione eccessivamente ridotta e potrebbero anch'essi subire dei sensibili cali di performance dovuti a frammentazione.

Per rispondere in maniera semplice a questi inconvenienti si è preferito ricorrere ad un compromesso tra i due estremi appena esposti: si stabilisce che all'interno di ogni gruppo esista un nodo che svolge la funzione di coordinatore dei vari membri e si preoccupi di raccogliere da questi ogni sorta di informazione utile per poter sintetizzare una serie di proprietà di più alto livello che descrivono lo stato complessivo del gruppo, in accordo a quanto esposto nel paragrafo precedente. Si impone che un qualsiasi veicolo deve essere all'interno della portata di un leader per potersi unire al suo gruppo. Da questa scelta consegue il fatto che il leader di un gruppo è in grado di comunicare direttamente con qualsiasi suo membro e viceversa, tuttavia non viene in

alcun modo garantito che i diversi nodi riescano sempre a scambiarsi informazioni. Questa limitazione è comunque perfettamente tollerabile: nonostante il sistema sviluppato non ricorra mai ad un contatto diretto tra due nodi membri questo resta comunque facile da realizzare servendosi del leader del gruppo come intermediario tra i due pari.

Il fatto che un veicolo si trovi nel raggio di copertura di un group leader non è una condizione sufficiente per stabilire la sua ammissibilità come membro del gruppo. Si consideri ad esempio una carreggiata percorribile in due direzioni: due veicoli che si trovano a percorrere la strada con sensi di marcia opposti arrivano a trovarsi molto vicini nel momento in cui si incrociano, tuttavia, per i requisiti che abbiamo fissato, non è possibile considerarli come potenziali membri di uno stesso gruppo. Per come è stato definito, un plotone viene sempre composto da veicoli che condividono una stessa direzione, cioè che in un dato istante si stanno dirigendo in maniera pseudo-solidale verso uno stesso punto. Una RSU posta a gestione di un incrocio stradale o di un qualsiasi snodo sensibile può fare affidamento a questa fondamentale caratteristica dei gruppi per differenziare il traffico in avvicinamento a seconda della sua provenienza.

Per regolare la formazione dei gruppi di veicoli con la procedura già suggerita in precedenza viene sfruttata la definizione di direzione introdotta nel primo paragrafo di questo capitolo: si uniscono tra loro nodi che riescono a comunicare in maniera diretta con un loro pari eletto al ruolo di leader e che possiedono lo stesso valore di direzione. Ogni plotone contribuisce in questo modo a raccogliere e trasmettere delle informazioni relative esclusivamente allo stato del traffico nel senso di marcia che sta percorrendo.

I dispositivi interessati a raccogliere nel tempo una base di conoscenza sulle proprietà del flusso locale di autoveicoli dispongono di una lista predefinita di valori di direzione (misurati in gradi d'arco) per le quali si aspettano di ricevere informazioni provenienti da particolari nodi eletti al ruolo di leader di gruppo. I dati che ogni gruppo trasmette alle RSU riassumono lo stato dei veicoli di cui è composto e possono includere nozioni sul numero di veicoli appartenenti al plotone, la loro velocità media, minima o massima, la densità media rilevata dai singoli membri, l'estensione massima del gruppo o qualsiasi altra caratteristica che contribuisca a delineare le condizioni del traffico stradale in quel punto.

Per realizzare concretamente questo meccanismo, ogni RSU trasmette un segnale di beacon ad intervalli regolari e ravvicinati il quale descrive la sua identità e le

direzioni alle quali è interessata. Qualsiasi nodo inattivo che riceve tale segnale confronta la lista dei valori indicati con la propria direzione media attuale e, se quest'ultima risulta conformante, si attiva per quel senso di marcia. È necessario quindi che i nodi attivi giungano al più presto alla formazione di un plotone secondo le modalità sopra descritte ed inizino a raccogliere dati da sintetizzare e da consegnare in seguito alle unità infrastrutturali. A tale scopo essi devono eleggere un leader che si preoccupi di coordinare i diversi membri del gruppo. Siccome plotoni con un maggior numero di membri sono generalmente in grado di raccogliere ed integrare un quantitativo di dati superiore e rappresentano più concretamente lo stato del traffico se la densità è elevata, è importante assicurarsi la formazione di gruppi di grandi dimensioni. Questo è reso possibile guidando la scelta del coordinatore in modo da favorire l'elezione di quei veicoli che vedono all'interno della loro portata il più alto numero di nodi conformanti alla direzione richiesta dalla RSU.

Se si assume l'ipotesi che in assenza di una RSU i nodi si muovano in uno stato di completa inattività, questi si ritroveranno privi di una qualsiasi forma di conoscenza sull'ambiente circostante nel momento in cui saranno chiamati ad individuare un nodo adeguato per il ruolo di leadership, ovvero quello in grado di comunicare con il maggior numero di veicoli. Sotto tali condizioni una strategia piuttosto semplice con la quale essi possono accordarsi è formata da una prima fase nella quale i diversi partecipanti segnalano la propria presenza e la propria identità attraverso un messaggio multicast indirizzato a tutti i nodi che condividono la stessa direzione prevista per il gruppo che intendono formare, seguita da una sorta di asta nella quale i diversi veicoli si propongono come candidati al ruolo di coordinatore: ciascun nodo che intende partecipare all'elezione invia un messaggio ai nodi circostanti indicando la propria identità, la direzione per la quale intende formare il gruppo ed il numero di veicoli che esso è in grado di raggiungere. Superate queste due fasi, si ottiene idealmente una conoscenza distribuita dei veicoli circostanti e del loro vicinato ed è possibile individuare un leader idoneo per il gruppo selezionando quel nodo che risulta capace di coordinare un maggior numero di membri³. Questa tecnica di elezione risulta piuttosto semplice da realizzare ma comporta lo scambio di un quantitativo di messaggi considerevole tra nodi che tendono a concentrarsi in un punto e può presentare degli svantaggi nel momento in cui la densità dei veicoli in avvicinamento alla RSU supera una certa soglia critica.

³ In caso di parità tra più veicoli ed in mancanza di ulteriori esigenze, si può scegliere in maniera indifferente uno qualsiasi tra essi.

Se fosse invece possibile assumere che i nodi non ancora attivati da un dispositivo infrastrutturale possano comunque scambiarsi occasionalmente dei semplici messaggi allo scopo di continuare a monitorare le caratteristiche dell'ambiente circostante, essi riuscirebbero a sfruttare tale conoscenza per individuare un leader in maniera più semplice e con uno scambio di dati decisamente più contenuto. Anche questa soluzione non è però esente da difetti visto che il meccanismo utilizzato per mantenere coscienza dei nodi vicini richiede necessariamente l'invio continuo di messaggi broadcast, anche nei momenti in cui tali informazioni non sono strettamente necessarie.

Queste due diverse tecniche di raggruppamento si differenziano dal fatto di adottare dei processi di tipo reattivo o proattivo per arrivare a individuare il giusto candidato al ruolo di leadership. Nel protocollo decentralizzato sono entrambe realizzate in maniera indipendente e verranno approfondite in seguito.

Indipendentemente dal metodo utilizzato per l'elezione di un nodo leader, una volta attivato questo deve preoccuparsi di controllare continuamente le condizioni dei propri nodi membri per assicurarsi della loro presenza e raccogliere da loro informazioni rilevanti per il protocollo. Una modalità particolarmente semplice e di facile implementazione per realizzare questa funzionalità è l'utilizzo di regolari beacon inviati dal leader ai propri membri che sono tenuti a rispondere alle richieste inviando un messaggio contenente la loro identità ed il loro stato attuale. In questo modo il responsabile di un gruppo può continuare a verificare nel tempo la presenza dei propri sottoposti e contemporaneamente ricevere da essi le informazioni necessarie a svolgere il proprio operato.

Se un veicolo di un gruppo, incluso il leader, rileva che la propria direzione non è più conforme a quella prevista dalla RSU deve necessariamente uscire dal plotone e ritornare ad uno stato di quiescenza. Ovviamente, se è il leader stesso a cambiare percorso o a perdere per qualche motivo la propria connettività, conviene che i nodi rimasti orfani procedano al più presto all'elezione di un nuovo gestore. Se il leader di un plotone non riceve alcuna risposta da parte dei suoi membri può comunque mantenere il suo stato di coordinatore ed attendere eventualmente che altri nodi si uniscano ad esso in futuro. Sono dunque perfettamente ammissibili dei gruppi composti da un solo individuo, condizione che può manifestarsi con frequenza in caso di un flusso di traffico contenuto.

Si prevede infine che qualsiasi nodo inattivo che avverte la presenza di un gruppo dotato di direzione compatibile con quella che sta attualmente percorrendo possa effettuare l'accesso a quel plotone, a patto che venga garantita la visibilità del

suo leader. Un singolo nodo membro può autonomamente uscire dal gruppo a cui appartiene nel momento in cui rileva che la sua distanza dalla RSU è in aumento e che quindi tale veicolo ha già superato l'incrocio stradale che si vuole monitorare. Inoltre il leader di un gruppo nel quale è il solo membro può in qualsiasi momento decidere di unirsi ad un altro plotone conformante in qualità di semplice membro.

Questi accorgimenti si rendono necessari dal momento che ci troviamo ad operare in scenari dove la mobilità dei singoli nodi è piuttosto elevata e non vi è alcuna garanzia che una volta conclusa la fase di formazione di un gruppo la sua conformazione rimanga invariata ed i diversi veicoli che lo compongono continuino a mantenere nel tempo la stessa posizione relativa e una direzione sempre conforme a quella prevista. Un plotone di veicoli è dunque un'entità dinamica ed in continua evoluzione, in cui gli elementi che lo costituiscono possono unirsi, aggiungersi e disgregarsi a seconda del percorso che stanno seguendo e della propria posizione corrente.

Nell'intervallo di tempo che va dall'avvenuta formazione di un gruppo alla sua chiusura, il leader si preoccupa di inviare alla RSU, in istanti più o meno equidistanti, un aggiornamento sul proprio stato e sulle statistiche raccolte fino a quel momento dal plotone di propria competenza. In questo modo le unità dell'infrastruttura stradale possono godere di informazioni aggiornate sui plotoni attivi in un dato momento e possono oltretutto disporre di una storia di campionamenti per ciascuno di essi. Questa serie di informazioni può risultare talvolta indispensabile per poter applicare delle logiche di gestione del traffico sofisticate ed in grado di differenziarsi in base all'evoluzione delle condizioni del traffico nel tempo.

Il ciclo di vita di un gruppo termina nel momento in cui il plotone raggiunge una distanza sufficientemente piccola dalla RSU ed è necessario procedere alla sua chiusura dal momento che, superato l'incrocio stradale, i diversi veicoli del gruppo inizieranno a percorrere strade differenti ed il concetto di gruppo per come è stato definito viene del tutto a mancare. Le modalità con cui questo evento viene innescato possono essere molteplici: si potrebbe individuare un punto centrale per il gruppo considerando le posizioni dei diversi membri ed utilizzarlo come riferimento. Questo comporterebbe però un problema aggiuntivo se si considera che non vi è sicurezza che il leader del gruppo venga posizionato al centro o comunque mantenga nel tempo una posizione centrale e potrebbe quindi uscire dall'incrocio prima di tale ipotetico baricentro. Per non introdurre ulteriori complessità si è deciso di adottare una soluzione decisamente semplice ma in grado di fornire comunque buoni risultati: si stabilisce che il leader monitori costantemente la propria distanza dalla RSU ed inneschi il processo di chiusura per il suo gruppo non appena questa scende sotto una soglia configurabile.

Quando questa condizione viene soddisfatta il gruppo ha di fatto concluso il suo compito ed i veicoli che lo compongono possono tornare ad uno stato di quiescenza, evitando ovviamente di riattivarsi fintanto che il dispositivo che hanno appena servito rimane nella loro portata. Anche la RSU viene informata dell'avvenuta terminazione del plotone, in modo che possa adeguatamente tenere traccia dei gruppi che terminano correttamente il proprio ciclo di vita e differenziare questi ultimi da altri gruppi che possono invece cessare le comunicazioni in maniera anomala.

2.3 Protocollo centralizzato

A differenza della strategia descritta in precedenza basata sul raggruppamento dei veicoli e sulla decentralizzazione della raccolta dei dati, questa seconda strategia si pone come obiettivo quello di centralizzare la raccolta e la manipolazione dei dati all'interno della RSU. Il processo di data fusion descritto in precedenza è integrato all'interno del dispositivo infrastrutturale bypassando la necessità di raggruppare i veicoli in plotoni. Facendo riferimento alla Figura 2.2, il livello intermedio composto dai gruppi viene rimosso ed i singoli nodi comunicano direttamente con la RSU.

Uno dei motivi principali per lo sviluppo di questo protocollo è stata la ricerca di migliori prestazioni in condizioni di bassa penetrazione delle tecnologie di comunicazione V2X, circostanza che sarà probabilmente dominante nel mondo reale nel prossimo futuro, almeno fino a quando la maggior parte dei produttori di autovetture non avrà implementato queste capacità sulle proprie automobili. La bassa diffusione delle tecnologie utilizzate nei sistemi sviluppati ha portato a focalizzarsi sulla raccolta di informazioni sul flusso del traffico che fossero indipendenti dal numero di vetture coinvolte nella loro generazione, come ad esempio la velocità di una veicolo o il tempo da loro impiegato per superare un incrocio stradale. Queste grandezze, a differenza del numero di autovetture attualmente in attesa di attraversare l'intersezione che hanno partecipato al protocollo, non dipendono dal grado di penetrazione dei sistemi V2X e presentano al sistema di controllo della logica semaforica una approssimazione della realtà più fedele, dal momento che è priva di assunzioni sulla diffusione delle tecnologie di trasmissione fra i veicoli che hanno preso parte alla raccolta delle informazioni.

Il protocollo decentralizzato discusso in precedenza ha lo svantaggio di essere abbastanza complesso e di richiedere un tempo di setup precedente allo scambio delle informazioni non trascurabile, derivante dalla necessità di raggruppare i veicoli in gruppi prima di poter procedere alla raccolta dei dati. La riduzione della penetrazione delle tecnologie di comunicazione car-to-car comporta una diminuzione del numero

medio di veicoli presenti all'interno dei singoli gruppi, fino ad arrivare al limite di plotoni composti da una sola entità. Tenendo conto di questa limitazione è stato scelto di rimuovere completamente il concetto di raggruppamento di veicoli nello sviluppo di questo protocollo, eliminando interamente l'overhead per la loro formazione. Questa scelta ha come effetto negativo il possibile incremento del traffico generato dai veicoli nel caso opposto a quello preso in esame, alta penetrazione delle tecnologie V2X, ed è stato pertanto necessario ripensare il modo con cui le informazioni sono raccolte: tale responsabilità è stata spostata dai veicoli eletti al ruolo di coordinatore di un gruppo alla RSU. Ciò ha permesso di semplificare notevolmente il protocollo di comunicazione e la logica necessaria all'interno dei veicoli che ora sono incaricati esclusivamente di raccogliere informazioni sul proprio stato e di recapitarle ai dispositivi infrastrutturali interessati a riceverle.

Un effetto collaterale di questo cambio di strategia è una maggiore quantità di informazioni raccolte dalla RSU sullo stato dell'incrocio, che deve essere quindi in grado di processare in tempo reale una grande mole di dati per fornire all'euristica di controllo dalla politica semaforica i dati di alto livello su cui prendere decisioni. Questo aspetto è però bilanciato da una maggiore semplicità di sviluppo di nuove logiche più avanzate che richiedono nuove o maggiori informazioni sullo stato dell'intersezione per poter funzionare. È infatti più semplice modificare le unità infrastrutturali in tempi ristretti e senza causare incompatibilità fra differenti versioni dei protocolli, rispetto ai singoli dispositivi di comunicazione installati all'interno di ogni veicolo. Ad esempio, la creazione di una logica di controllo che utilizzi il tempo necessario ad un veicolo per attraversare un incrocio come metrica di riferimento per effettuare decisioni risulta di facile deployment nel protocollo centralizzato: è infatti sufficiente che la RSU conservi l'istante nel quale ha ricevuto il primo messaggio da parte di un veicolo e lo confronti con l'istante nel quale l'ultimo messaggio è stato ricevuto per avere una stima del tempo impiegato per superare l'intersezione, senza la necessità di alcuna modifica al protocollo di comunicazione o ai dispositivi installati all'interno dei veicoli. Nel caso del protocollo decentralizzato una simile logica potrebbe essere implementata accettando una peggiore approssimazione considerando il tempo impiegato da un gruppo per attraversare un incrocio oppure tramite una modifica del protocollo di comunicazione, con la conseguente necessità di aggiornare i dispositivi installati su ogni nodo mobile, in modo che questa informazione sia tra quelle inviate alla RSU dal leader di un plotone.

L'architettura centralizzata di questo protocollo aumenta la flessibilità del sistema che può essere adattato a cambiamenti anche rilevanti nelle funzionalità richieste tramite la modifica delle sole unità infrastrutturali. La centralizzazione di un gran numero di funzionalità all'interno di uno stesso punto, benché porti ai vantaggi descritti, comporta anche una serie di problematiche che vanno gestite adeguatamente, in particolare la scalabilità e la complessità dei nodi che compongono l'infrastruttura.

In questo protocollo, in modo simile a quanto avviene per la soluzione decentralizzata descritta in precedenza, un dispositivo interessato a raccogliere dati sull'evoluzione del traffico nel tempo dispone di una lista di valori di direzioni, espressi in gradi d'arco, che rappresentano le arterie in ingresso all'intersezione da lui controllata per le quali è interessato a ricevere informazioni dai veicoli. I dati che vengono trasmessi dai veicoli riassumono il loro stato ed includono l'identità, la posizione, la direzione nella quale si stanno muovendo e la velocità attuale e media.

In fase di progettazione è stato deciso che per poter partecipare a questo protocollo i nodi devono essere all'interno del raggio di copertura dei messaggi inviati dalla RSU, in quanto non si è previsto l'utilizzo di un protocollo di routing che possa mettere in comunicazione due nodi che non hanno visibilità diretta fra di loro. Questa caratteristica limita la quantità di veicoli che possono prendere parte alla raccolta dei dati a quelli presenti all'interno del breve tratto precedente allo snodo stradale, ma sono stati ritenuti comunque sufficienti alla realizzazione una buona stima sul flusso del traffico. Tenendo conto dell'elevata mobilità dei nodi e delle innumerevoli condizioni ambientali che possono interferire o variare la portata dei segnali, il protocollo deve accettare la possibilità che una parte dei dati venga persa o che il raggio di copertura del segnale possa variare a seguito di variazioni delle condizioni fisiche.

La logica di raccolta dei dati è realizzata da questo protocollo tramite l'invio, ad intervalli regolari, di un messaggio di beacon da parte dei dispositivi infrastrutturali nel quale è specificata l'identità ed a quali direzioni si è interessati monitorare. Ogni nodo che riceve questo segnale confronta la propria direzione attuale con quella indicata ed in caso risulti conformante risponde al messaggio con le informazioni aggiornate sul proprio stato. In seguito a questo scambio di messaggi, il nodo si rimette in attesa di ricevere un altro segnale di beacon secondo una logica di funzionamento puramente reattiva, subordinata alla ricezione di un segnale di richiesta dati da parte di un dispositivo dell'infrastruttura. A differenza del segnale di beacon utilizzato nel protocollo decentralizzato che contiene la lista di tutte le direzioni alla quale la RSU è interessata, in questo protocollo i beacon contengono una singola direzione per limitare il numero di veicoli che possono rispondere contemporaneamente ed è quindi

necessario esplorare la lista delle direzioni in modo ciclico con segnali successivi per ottenere informazioni aggiornate per ogni strada in ingresso all'incrocio.

I nodi continuano a rispondere a beacon successivi di una RSU fintanto che la loro direzione attuale rimane conformante con una di quelle di interesse o fino al raggiungimento di una distanza sufficientemente piccola dal centro dell'incrocio. Nel caso un veicolo abbia superato la soglia specificata alla ricezione di un messaggio di beacon invia un'ultima risposta alla RSU indicando la fine della propria partecipazione alla raccolta dei dati. Questa condizione è utilizzata per evitare che i nodi che sono all'interno dell'intersezione e che sono in procinto di superarlo possano inviare informazioni falsate. Occorre inoltre garantire che un nodo, dopo aver superato l'incrocio, non continui ad inviare dati in risposta ai messaggi di richiesta inviati dalla RSU. Nel caso il dispositivo infrastrutturale necessiti di informazioni anche sulla condizione del traffico nei tratti immediatamente successivi allo snodo stradale che deve monitorare quest'ultima condizione può essere rimossa, a patto che sia possibile discriminare fra i nodi che sono ancora in attesa e quelli hanno già attraversato l'intersezione.

2.4 Classi di veicoli

Nei casi d'uso reali i molteplici dispositivi preposti alla comunicazione fra i nodi e gli strumenti utilizzati per rilevare le grandezze fisiche di interesse sono estremamente eterogenei tra loro e possano fornire risposte diverse a fronte di identici stimoli dati dall'ambiente. Più concretamente, si prevede che i diversi sensori utilizzati per registrare posizione, direzione e velocità di un veicolo si comportino con una precisione ed un'accuratezza diverse rispetto a quelle di analoghi sensori presenti in veicoli differenti. Anche gli stessi componenti per la trasmissione dei dati, pur ricorrendo allo stesso standard per le comunicazioni wireless, potranno reagire con diversi tempi di latenza e performance leggermente dissimili e raggi di copertura di elevata variabilità.

Occorre inoltre notare che gli studi su qualsiasi tipo di tecnologia VANET partono dal presupposto che esista almeno una certa percentuale di veicoli effettivamente dotata di sistemi in grado di rilevare la propria posizione e di comunicare con altri nodi vicini. Nonostante i sensori dedicati al posizionamento siano ormai molto diffusi grazie all'enorme successo di soluzioni di navigazione assistita, i dispositivi per la comunicazione V2X hanno per il momento una penetrazione di mercato praticamente nulla e gli investimenti da parte delle principali case automobilistiche risultano ancora contenuti e limitati prevalentemente a prototipi, sperimentazioni locali e ricerca.

Nell'attesa che questa strumentazione raggiunga un adeguato livello di maturità è possibile ipotizzare l'utilizzo dei già citati e diffusissimi smartdevice come interfacce di comunicazione e semplici apparecchiature di geolocalizzazione. Ci si aspetta tuttavia che le prestazioni fornite da tali dispositivi siano peggiori in termini di accuratezza rispetto a quelle ottenibili dalle soluzioni specifiche appena suggerite.

Naturalmente, le prestazioni dei singoli componenti responsabili del monitoraggio dello stato del veicolo e della comunicazione tra i diversi nodi incidono sulla resa complessiva dei protocolli sviluppati e sulla qualità dei dati ricevuti dalle RSU. Per tenere conto di questa variabilità e per poter studiare in che modo e con quale rapporto le risposte del sistema variano a seconda dei sensori e delle interfacce utilizzate sono state previste tre diverse classi di veicoli, differenziate in base alle loro capacità ed all'insieme dei dispositivi offerti:

Shadow vehicle Veicoli che non sono dotati di un'interfaccia di comunicazione wireless e risultano pertanto invisibili al sistema ed ai protocolli. Pur disponendo di eventuali sensori utili per ricavare la loro posizione, velocità e direzione, non riescono a contribuire alla creazione e al mantenimento di informazioni utili sullo stato del traffico. Si tratta del tipo di autovetture oggi principalmente in uso.

Mid-class vehicle Veicoli che non dispongono di un'interfaccia di comunicazione nativa, ma nei quali uno smartdevice personale dell'utente sopperisce a tale mancanza. Questo dispositivo è in grado di stimare la posizione del veicolo e di comunicare con altri nodi purché dotati di analoghe interfacce. Data la natura dei dispositivi si presuppone che la precisione degli strumenti e la portata dei segnali scambiati con altri veicoli siano peggiori rispetto a quelle offerte da vetture di classe full.

Full-class vehicle Veicoli dotati di sensori di geolocalizzazione di precisione elevata e con sistemi di comunicazione car-to-car integrati e di buona portata. Questa classe risulta essere ideale per i protocolli sviluppati e non soffre delle limitazioni presenti nei mid-class vehicle. Attualmente la diffusione di queste tecnologie nelle autovetture è praticamente nulla.

Capitolo 3

Architettura del sistema

In questo capitolo si vogliono presentare i dettagli sull'architettura sviluppata per realizzare concretamente il sistema di Intelligent Transportation System (ITS) presentato in precedenza con particolare attenzione alla logica di comportamento dei singoli protocolli, ai messaggi che essi scambiano tra loro ed alle modalità di funzionamento dei componenti basilari del sistema.

Prima di esplorare nel dettaglio i diversi aspetti architetturali della soluzione realizzata, verranno esposti i principali strumenti adottati per lo sviluppo e gli ambienti di simulazione impiegati per l'elaborazione ed il collaudo del lavoro eseguito. Si rimanda la lettura al capitolo 5 per qualsiasi informazione relativa alle modalità di esecuzione e di raccolta dei risultati sperimentali o agli specifici parametri di simulazione utilizzati nei diversi test.

3.1 Ambiente di simulazione

Lo sviluppo di nuove tecnologie software e hardware ricorre sempre più frequentemente all'utilizzo di appositi programmi dedicati alla simulazione degli specifici scenari di utilizzo e dei casi d'uso previsti per il particolare dominio applicativo. Tali strumenti permettono di studiare nel dettaglio il comportamento e la risposta del sistema in base a molteplici stimoli in ingresso. A tale scopo il progettista, dopo aver definito e codificato una versione sperimentale della logica di comportamento dei vari elementi costitutivi dell'applicazione in esame, ne effettua il deploy in un ambiente software che si occupa di simulare in maniera più o meno realistica lo scenario reale per il quale il sistema è stato concepito.

Questa modalità di progettazione si contrappone nettamente a quella più tradizionale che prevede una diretta sperimentazione nel reale ambiente di utilizzo che, specialmente nel caso di sviluppo di applicativi hardware, risulta considerevolmente più costosa sia in termini di tempo che di risorse necessarie. Nonostante l'emergere di soluzioni relativamente economiche ed accessibili ad un'utenza sempre più vasta

[Sad11; SK10], la progettazione di sistemi complessi, destinati ad un pubblico vasto ed a scenari di utilizzo differenziati non può prescindere dall'utilizzo di specifici ambienti di simulazione. Esistono inoltre casi nei quali la sperimentazione per mezzo di strumenti virtuali è l'unica opzione inizialmente possibile: ad esempio nel campo delle comunicazioni fra sistemi mobili è richiesta non solo la codifica degli specifici protocolli, algoritmi e framework che costituiscono il fondamento principale della tecnologia che si intende studiare, ma anche l'installazione di quest'ultima su un vasto numero di dispositivi, spesso eterogenei e dislocati nello spazio, per poter ottenere un riscontro significativo del loro comportamento. È evidente come già il semplice costo per l'acquisto o per la produzione dell'hardware necessario a tale sperimentazione risulti proibitivo per la maggior parte dei progettisti; anche il tempo necessario per la raccolta dei risultati nei diversi test, la ricodifica dei sorgenti in caso di errori o modifiche, il successivo aggiornamento di molteplici dispositivi e la nuova messa a punto dell'ambiente di sperimentazione comportano un ciclo di sviluppo estremamente lento e dispendioso.

Per i motivi appena esposti l'implementazione e la sperimentazione dell'architettura di ITS introdotta nel capitolo precedente sono stati realizzati all'interno di iTETRIS, un sistema di simulazione di mobilità veicolare e di comunicazione di rete finanziato dalla commissione europea [Ith14]. Questa piattaforma di simulazione è stata sviluppata appositamente per consentire la valutazione del design di sistemi di comunicazione veicolare e dell'impatto di politiche di gestione del traffico in ambienti realistici ed accurati di larga scala. Grazie alla natura open source di questa piattaforma è inoltre possibile studiare e modificare direttamente il codice sorgente intervenendo in prima persona per implementare nuove funzionalità e modelli sperimentali e per personalizzare il comportamento della piattaforma secondo le particolari esigenze di simulazione.

3.1.1 Architettura complessiva di iTETRIS

iTETRIS è una piattaforma di simulazione di comunicazioni veicolari che permette la definizione di scenari di traffico veicolare e di rete e la loro simulazione attraverso l'uso degli strumenti integrati per la simulazione del traffico di rete wireless, NS-3 [NSh14], e del traffico veicolare, SUMO [Kra12]. Questo sistema è stato sviluppato per valutare le performance delle strategie di gestione del traffico e per migliorare l'efficienza della rete stradale attraverso una simulazione realistica di ITS. Questa piattaforma è di grande importanza per lo sviluppo di ITS in quanto, prima della loro introduzione su larga scala nell'ambiente reale, occorre dimostrare quali

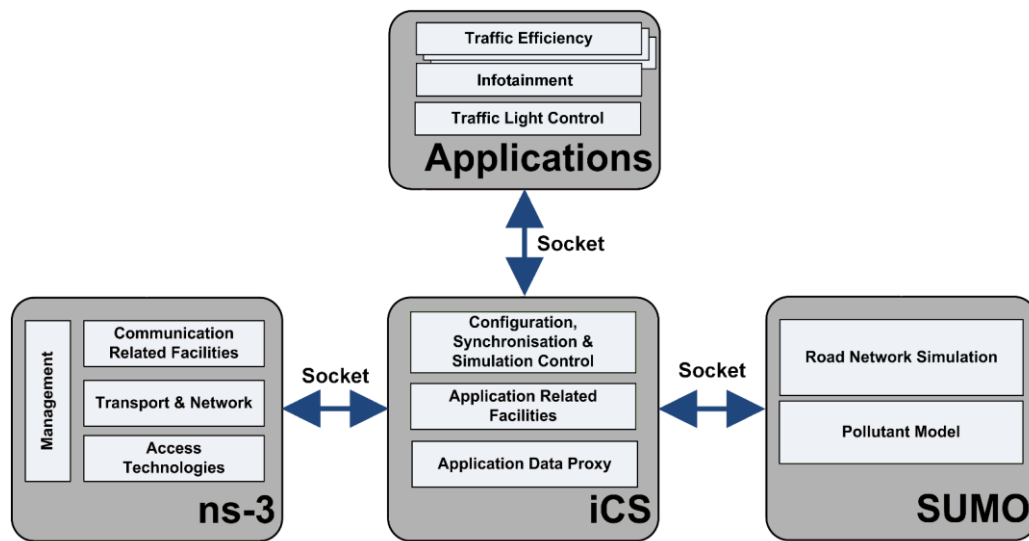


Figura 3.1: Architettura di alto livello di iTETRIS.

benefici si possono ottenere tramite l'utilizzo di queste soluzioni in scenari urbani realistici. Gli scenari simulati sono caratterizzati da un grande numero di veicoli distribuiti su un'area estesa dei quali viene studiato il comportamento per un determinato periodo di tempo.

La combinazione di questi due simulatori permette di valutare degli effetti dello scambio di informazioni fra veicoli sulla loro mobilità. In particolare, lo scambio di messaggi fra veicoli può portare a cambi di percorso; allo stesso tempo il movimento dei nodi ha un grande impatto sui dispositivi di trasmissione wireless e quindi sui protocolli di comunicazione utilizzati.

iTETRIS si compone di un terzo componente, chiamato iCS (iTETRIS Control System) che si occupa di coordinare i due simulatori in tempo reale e di fornire una interfaccia per consentire lo sviluppo di applicazioni. Tutte le interazioni fra i vari moduli passano attraverso questo componente centrale, come mostrato nella Figura 3.1. La modularità di questa architettura permette lo sviluppo indipendente dei vari componenti dalle rispettive comunità di sviluppatori facilitando la manutenibilità e l'estensione del sistema.

Per facilitare lo sviluppo di applicazioni iCS mette a disposizione dello sviluppatore una API che permette di controllare sia lo spostamento dei veicoli, che le loro comunicazioni, garantendo un controllo completo sulle azioni dei nodi mobili. È inoltre possibile comandare variazioni della logica di controllo degli impianti semaforici.

Queste possibilità permettono di sviluppare un grande numero di applicazioni differenti, dalla raccolta di informazioni sul flusso del traffico alla sicurezza attiva fra veicoli, o a servizi di infotainment. Le applicazioni che possono essere simulate si avvalgono delle funzionalità offerte da iCS, ma sono esterne alla piattaforma consentendo allo sviluppatore di scegliere il linguaggio di programmazione più opportuno per la propria soluzione.

Nei seguenti paragrafi verranno introdotti i due simulatori integrati nella piattaforma, spiegato il funzionamento di iCS ed illustrata l'interfaccia offerta per lo sviluppo di applicazioni.

3.1.2 SUMO

Un qualsiasi progetto di ricerca in ambito VANET non può prescindere dall'utilizzo di uno strumento in grado di simulare la mobilità di un insieme di veicoli all'interno di un reticolo stradale: le caratteristiche e la qualità delle trasmissioni wireless tra più nodi dipendono fortemente dalla posizione che essi assumono in ogni preciso istante di simulazione ed è quindi indispensabile poter conoscere nel dettaglio l'intero percorso seguito da qualsiasi veicolo simulato nell'ambiente. Normalmente queste informazioni sono contenute in una traccia di mobilità, che consiste in una formale descrizione delle coordinate spaziali e della velocità assunte dai nodi in ogni momento, ottenuta separatamente prima dell'inizio della simulazione dei protocolli di trasmissione. A differenza delle normali piattaforme di simulazione nelle quali la traccia di mobilità è statica e non può essere modificata in base a decisioni compiute dalle logiche che si stanno analizzando, grazie all'utilizzo di iTETRIS si può interagire dinamicamente sul flusso dei veicoli, in seguito alle azioni richieste dal protocollo in sperimentazione. Questo consente una migliore valutazione delle prestazioni del sistema in un ambiente più simile a quello reale.

Il simulatore di mobilità integrato in iTETRIS è SUMO (Simulation of Urban Mobility), un simulatore urbano open-source ad eventi discreti in grado di imitare in maniera estremamente realistica il comportamento di diversi tipi di automezzi che si spostano all'interno di una rete stradale; quest'ultima può essere interamente costruita dall'utente oppure ottenuta a partire da mappe realmente esistenti. Per integrare questo simulatore nell'architettura complessiva è stata sfruttata l'interfaccia TraCI (Traffic Control Interface) integrata in SUMO che permette ad un programma esterno di accedere ad una simulazione mentre questa è in esecuzione. Questa interfaccia permette l'utilizzo di un modello di comunicazione client - server, nel quale iCS agisce come client generando comandi e richiedendo dati, e SUMO agisce come server eseguendo

i comandi ricevuti. In questo modo è possibile avviare e terminare simulazioni, ottenere i dati relativi ai veicoli (come posizione, velocità o direzione) ed ai dispositivi infrastrutturali (come lo stato delle luci di un semaforo), e modificare lo scenario di simulazione, alterando il percorso di un veicolo o lo stato di un semaforo. Le funzionalità messe a disposizione da questa interfaccia hanno permesso agli sviluppatori di iTETRIS di utilizzare questo simulatore senza la necessità di apportare modifiche, a differenza di quanto verrà illustrato nel prossimo paragrafo per l'integrazione della piattaforma NS-3.

Gli scenari urbani utilizzati come riferimento per i vari test illustrati nel capitolo 5 sono state costruite sfruttando delle rilevazioni reali effettuate in aree urbane strategiche della città di Bologna. Le zone coinvolte nel campionamento sono normalmente note per soffrire di problemi di congestione e riguardano un totale di 24 ore per ciascun scenario. Selezionando di volta in volta un opportuno intervallo temporale si può disporre di tutte le diverse tipologie di traffico che è possibile incontrare durante un'intera giornata: da un flusso estremamente denso e rallentato relativo alle principali ore di punta a quello ridotto e scorrevole tipico delle ore notturne. Per ulteriori dettagli sugli scenari utilizzati come ambienti di test e sui parametri di simulazione in essi utilizzati si consultino i paragrafi 5.1 e 5.3.

3.1.2 NS-3

Nell'architettura di iTETRIS il componente incaricato alla simulazione delle comunicazioni fra i dispositivi mobili e dell'infrastruttura è il simulatore NS-3. In questo simulatore sono implementate una serie di interfacce di comunicazione wireless, utilizzate dai nodi per trasmettere e ricevere dati, e gli stack protocollari richiesti per permettere la comunicazione tra loro. Per consentire l'integrazione all'interno di iTETRIS, NS-3 è stato esteso aggiungendo una serie di funzionalità, come illustrato nella Figura 3.2. Per ridurre la necessità di interscambio di dati tra iCS ed NS-3 e per mantenere una separazione delle funzionalità, tutti i componenti necessari per la gestione delle comunicazioni wireless fra i veicoli sono state implementate all'interno di NS-3. In particolare, per ottimizzare la simulazione di reti VANET è stato realizzato uno stack C2C (Car 2 Car) che contiene specifici livelli di trasporto e di rete dedicati alle comunicazioni fra veicoli. All'interno di questo stack è presente un sistema di indirizzi comune e generico chiamato C2CAddress, capace di supportare differenti formati di indirizzamento per permettere comunicazioni basate sull'identificativo dei nodi, oppure per abilitare il broadcast geografico, senza la necessità di utilizzare indirizzi IP [Ron13].

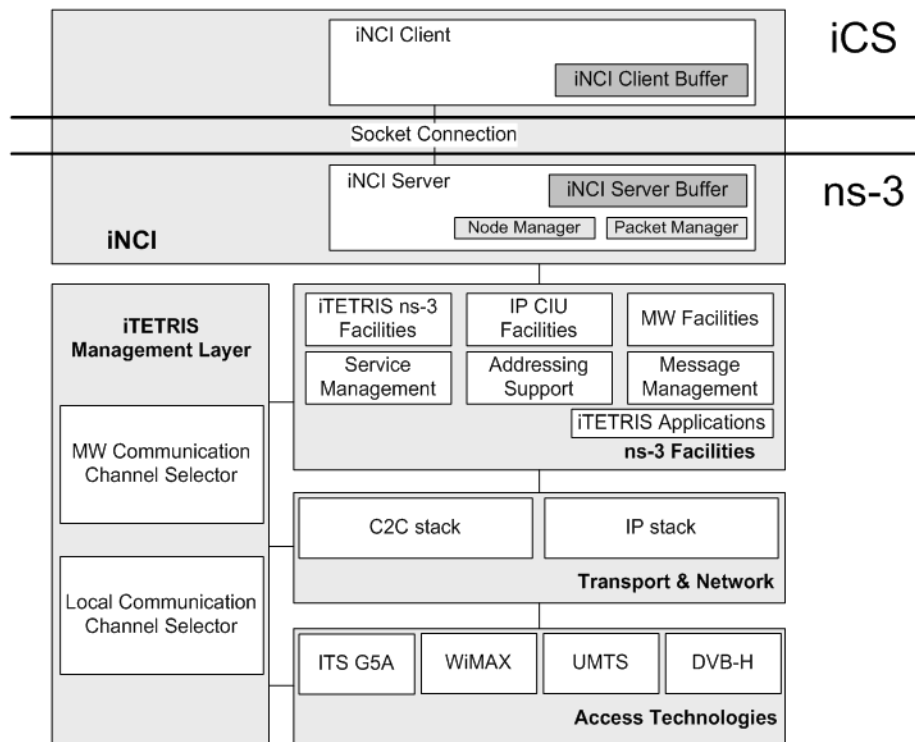


Figura 3.2: Componenti aggiunti all'architettura di NS-3 per permettere l'integrazione all'interno di iTETRIS.

A differenza di SUMO, la versione standard di NS-3 non fornisce nessuna interfaccia per l'interazione con moduli esterni quindi, per sopperire a questa mancanza, è stata creata un'interfaccia di comunicazione fra iCS ed NS-3, chiamata iTETRIS Network simulator Control Interface (iNCI), che si occupa di gestire le comunicazioni fra i due moduli. La logica implementativa di questa interfaccia è molto simile a quella utilizzata per le comunicazioni con SUMO: è basata su un modello di comunicazione client - server, nel quale iCS agisce come client controllando l'esecuzione di NS-3, che si occupa di eseguendo le azioni richieste agendo come server. L'implementazione del modulo iNCI all'interno di NS-3 si compone di due entità principali: un node manager ed un packet manager. Il gestore di nodi permette ad iCS la creazione dinamica di nuovi nodi e l'aggiornamento delle posizioni e delle velocità dei nodi esistenti ad ogni passo di simulazione. Il gestore dei pacchetti consente di schedulare l'invio di messaggi da parte di un nodo e di ottenere i messaggi ricevuti correttamente.

NS-3 è un simulatore ad eventi discreti open source finalizzato allo studio, allo sviluppo ed alla messa a punto di soluzioni e protocolli di comunicazione tra nodi e apparecchiature di rete. Mediante questo programma è possibile definire le proprietà

e le interfacce dei nodi presenti nell'ambiente, definire una topologia di rete che mette in comunicazione tali dispositivi, installare e configurare gli opportuni stack protocolari che si intendono utilizzare per fornire ai nodi i diversi servizi previsti dal modello ISO/OSI [Tan02] ed infine aggiungere un eventuale insieme di applicazioni che generano traffico all'interno della rete e processano i dati trasportati per mezzo di pacchetti. Per realizzare queste funzionalità vengono create e connesse tra loro numerose classi di oggetti che ripropongono il comportamento delle principali entità fisiche presenti nel mondo reale (nodi fissi o in movimento, mezzi per la trasmissione dei segnali, dispositivi di rete, edifici, etc.), codificano gli algoritmi ed i protocolli presenti in letteratura (stack internet, routing, applicazioni di rete, etc.) e simulano in maniera sufficientemente realistica l'effetto di alcuni fenomeni fisici di interesse (segnali elettrici ed elettromagnetici, propagazione, errori, consumi energetici, etc.). Per una descrizione completa delle funzionalità di NS-3 è possibile fare riferimento alla documentazione presente sul sito ufficiale del progetto [NSH14].

3.1.4 iTETRIS Control System

Come illustrato nella Figura 3.1 la parte centrale della piattaforma di simulazione è costituita da iCS. Questo modulo oltre a permettere l'interazione fra i due simulatori, come illustrato nei precedenti paragrafi, fornisce anche un'interfaccia utente per la creazione di applicazioni di gestione del traffico. Come mostrato in figura ogni interazione tra i componenti della piattaforma ha come origine o come destinazione questo modulo, è quindi di vitale importanza la sua ottimizzazione. L'interazione fra NS-3 e SUMO avviene sempre attraverso iCS, mai in modo diretto, a differenza di quanto precedentemente studiato da altre soluzioni che prevedevano l'accoppiamento diretto fra SUMO ed NS-2 (una versione precedente di NS-3) [KTW08]. Per ottenere un'interazione efficiente fra i vari componenti del sistema, durante lo sviluppo di iCS sono state valutate varie tecnologie alternative come CORBA e SOAP, scegliendo infine di utilizzare socket IP [Man09]. Nonostante questo meccanismo sia semplice e di basso livello, iCS può comunicare con gli altri blocchi dell'architettura attraverso protocolli sviluppati appositamente, velocizzando le simulazioni e riducendo al minimo la quantità di dati scambiati.

L'interazione fra iCS ed i due simulatori è ottenuta attraverso una comunicazione bidirezionale. iCS controlla l'esecuzione di SUMO per emulare il movimento degli veicoli secondo lo scenario stradale utilizzato e può inviare una serie di comandi per specificare le azioni da compiere sull'ambiente simulato con lo scopo di modificare il flusso del traffico. Allo stesso tempo iCS richiede a SUMO la posizione dei

veicoli per aggiornare gli altri moduli del sistema, nel caso la posizione sia variata nel tempo. Allo stesso modo iCS comanda la simulazione delle comunicazioni wireless in NS-3 schedulando i messaggi che devono essere inviati, aggiornando i nodi con le posizioni aggiornate ottenute da SUMO e recuperando i messaggi che sono stati ricevuti correttamente per poterli comunicare alle applicazioni utente che provvederanno a processarli.

Le principali funzioni di iCS sono quelle di attivare le differenti piattaforme, impostare l'ambiente di simulazione e di controllare e coordinare l'esecuzione della simulazione. Quando viene avviata una simulazione iCS provvede a leggere i propri file di configurazione e ad avviare e configurare correttamente i simulatori e le applicazioni. Al termine di questa fase di configurazione tutti i moduli sono stati inizializzati correttamente ed è possibile iniziare ad eseguire la simulazione. Il controllo e la coordinazione dei vari moduli della piattaforma durante l'esecuzione è di vitale importanza per ottenere risultati utili alla valutazione del comportamento del sistema studiato. È compito di iCS garantire che tutti i blocchi della piattaforma mantengano una sincronizzazione sia nel tempo che nello spazio: occorre infatti ricordare che sia i due simulatori sia le applicazioni, benché siano stati accorpati all'interno di questa singola piattaforma, sono programmi separati che eseguono in modo indipendente. La sincronizzazione temporale è indispensabile per assicurare che gli eventi eseguiti dai vari moduli seguano un ordine corretto, che le azioni svolte siano correttamente sincronizzate e correlate fra di loro e che il tempo di simulazione sia incrementato correttamente in ogni componente della piattaforma. La posizione dei nodi influenza in modo diretto le loro capacità di comunicazione con altri veicoli e dispositivi infrastrutturali ed influenza l'esecuzione dei protocolli applicativi, è pertanto indispensabile una opportuna sincronizzazione nello spazio i vari moduli del sistema.

Per ottenere questo comportamento iCS introduce una propria astrazione di nodo, che associa ad ogni veicolo di SUMO un nodo all'interno di NS-3 in modo che ogni dispositivo ITS simulato abbia una propria rappresentazione specifica all'interno di ogni blocco dell'architettura. Tramite questa astrazione è possibile coordinare correttamente le azioni che devono compiere i nodi ed è possibile aggiornarne correttamente le proprietà.

L'esecuzione di una simulazione in iTETRIS è suddivisa in step di sincronizzazione di durata fissa. Ogni step si compone di varie fasi come descritto in Figura 3.3: l'esecuzione inizia con una fase di esecuzione di NS-3 seguita dall'esecuzione di SUMO, da una fase nella quale sono eseguite le applicazioni ed termina con una fase dedicata alla preparazione del passo successivo. La durata predefinita di uno step è di

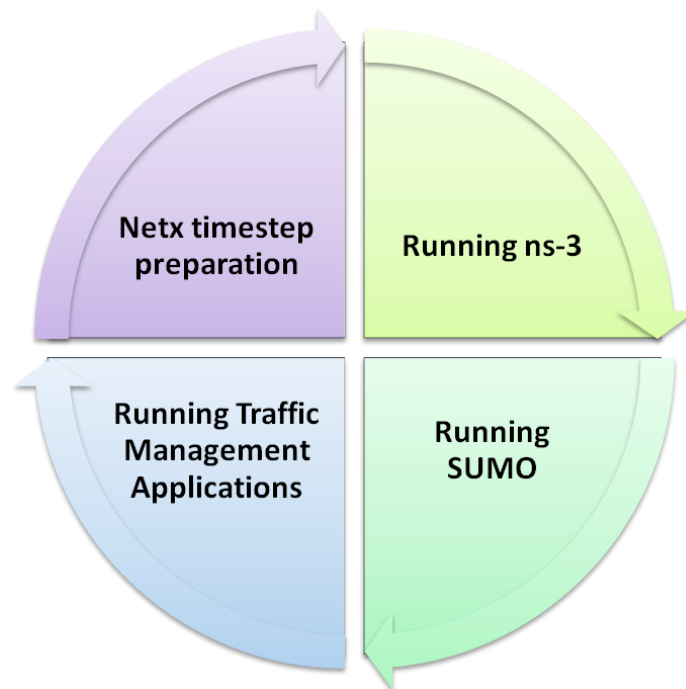


Figura 3.3: Sequenza di operazioni svolte da iCS ad ogni passo di simulazione.

un secondo, cioè per ogni ciclo SUMO ed NS-3 eseguono tutti gli eventi pianificati nell'intervallo di tempo di un secondo. Questo intervallo di tempo non va confuso con il periodo necessario all'esecuzione di questi eventi, infatti ci si riferisce al tempo interno alla simulazione, non a quello reale impiegato dal simulatore per simulare tali eventi. Per ottimizzare l'esecuzione dei protocolli sviluppati in questo progetto, il tempo di simulazione è stato ridotto e reso configurabile, con un valore minimo di un millisecondo (maggiori informazioni saranno presentate nel paragrafo 5.2).

L'interfaccia utente messa a disposizione dello sviluppatore da iCS permette di utilizzare più applicazioni contemporaneamente, che sono eseguite in successione nella terza fase di ogni step di simulazione. Queste applicazioni sono indipendenti fra di loro e sono collegate ad iCS attraverso una socket IP. All'avvio di una simulazione iCS si occupa del loro avvio e della loro configurazione attraverso le impostazioni specificate nei file di configurazione. L'utilizzo di un collegamento tramite socket lascia la massima libertà allo sviluppatore sia nella scelta del linguaggio di programmazione che nella loro realizzazione.

3.1.5 Applicazioni utente iTetris

Le interazioni fra le applicazioni ed iCS si basano su un meccanismo che utilizza subscription e result container. In generale una applicazione contiene algoritmi che utilizzano i dati ricevuti da iCS e restituiscono risultati che possono essere riutilizzati dalla stessa o da altre applicazioni, oppure scatenare azioni nella simulazione di rete o veicolare (ad esempio cambiare il percorso di una vettura a seguito di una notifica di un ingorgo).

Le sottoscrizioni rappresentano l'unico meccanismo con cui l'applicazione può comunicare con iCS: questo componente è utilizzato per ricevere ed inviare dati e per comandare l'esecuzione di azioni nei due simulatori. Una sottoscrizione può essere permanente oppure valere una volta sola, ad esempio la sottoscrizione per ricevere messaggi permette di ottenere dati fino a quando questa non viene esplicitamente cancellata dall'applicazione; diversamente la sottoscrizione che permette di schedulare l'invio di un messaggio attraverso NS-3 è valida una sola volta, anche se l'applicazione non la cancella esplicitamente iCS comanderà l'invio di un solo messaggio. Gli sviluppatori di iCS hanno messo a disposizione una serie di sottoscrizioni che soddisfano i casi d'uso più comuni ma c'è la possibilità di crearne di nuove per soddisfare gli specifici bisogni di una particolare applicazione. Purtroppo per poter sviluppare nuove sottoscrizioni occorre una conoscenza abbastanza approfondita del funzionamento interno di iCS limitando l'usabilità di questa funzione.

Fra le sottoscrizioni predefinite di maggiore interesse vi sono quelle per schedulare l'invio di messaggi, per ricevere i messaggi correttamente trasmessi, per comandare il cambiamento del percorso di un veicolo e per ottenere informazioni sullo stato dei nodi. Anche l'interazione predefinita fra applicazioni utenti differenti avviene tramite l'utilizzo di una particolare sottoscrizione che permette di accedere ai dati contenuti nel result container di un'altra applicazione. La comunicazione possibile attraverso questo meccanismo è abbastanza limitata, è infatti permesso il solo scambio dati, ma grazie alla grande libertà lasciata nello sviluppo delle applicazioni, è possibile implementare interazioni più complesse all'esterno di iCS, se necessario. In Figura 3.4 sono mostrati alcuni tipi di sottoscrizione.

I result container sono strutture dati all'interno di iCS che consentono di salvare informazioni generate dalla applicazione. Queste informazioni possono essere riutilizzate dall'applicazione stessa oppure possono servire per lo scambio di dati fra applicazioni diverse. I result container sono molto importanti nel caso più applicazioni

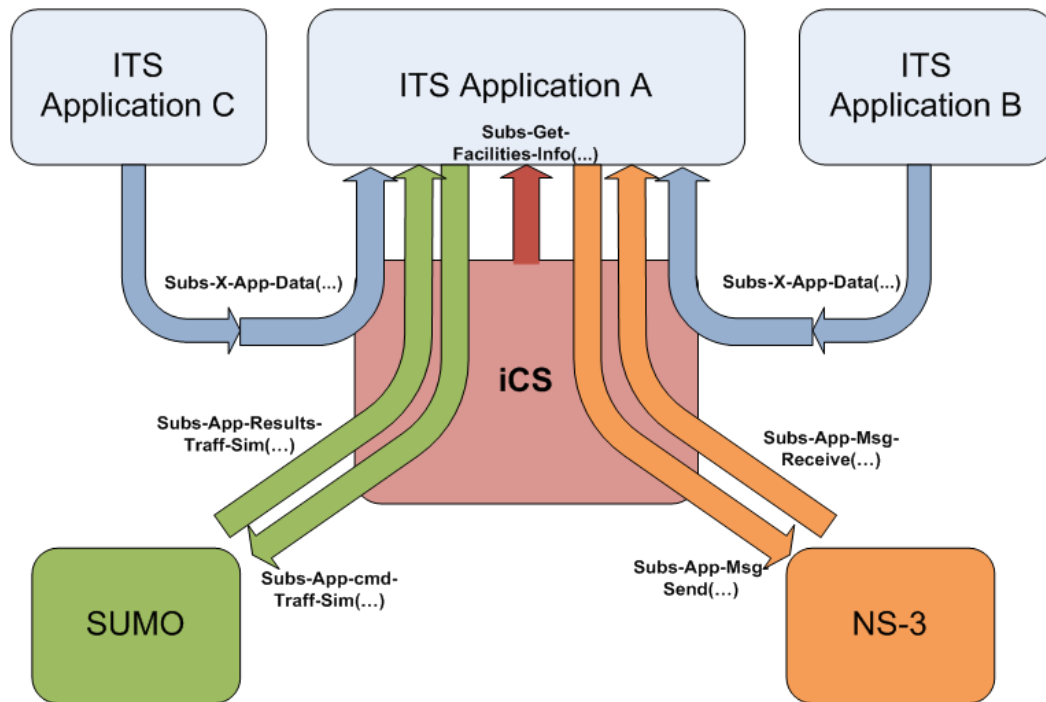


Figura 3.4: Alcune sottoscrizioni predefinite di iTETRIS.

necessitano di comunicare fra di loro, infatti sono l'unico strumento messo a disposizione da iCS per eseguire questo tipo di operazione.

Questo componente fa parte di iCS ma è sviluppato dal programmatore dell'applicazione in modo che sia possibile personalizzarlo secondo le proprie necessità. Purtroppo è necessario che lo sviluppatore conosca in modo approfondito il funzionamento interno di iCS per personalizzarne il comportamento. iTETRIS, come nel caso delle sottoscrizioni, fornisce alcuni result container predefiniti che coprono i casi d'uso più comuni, in particolare è presente una implementazione generica che permette all'applicazione di salvare e recuperare dati di vario tipo ed una implementazione vuota utile per quelle applicazioni che non sono interessate ad utilizzare questo componente.

Le possibilità offerte dai result container non solo limitate al solo salvataggio di dati, è possibile implementare funzionalità più avanzate: il suo posizionamento in iCS permette di avere accesso alle logiche interne della piattaforma e può quindi essere programmato per agire per conto di un'applicazione comandando azioni a seguito di particolari eventi. Ad esempio un result container potrebbe essere incaricato dall'applicazione di ricevere i messaggi che confermano la ricezione di una trasmissione inviata ad un altro nodo.

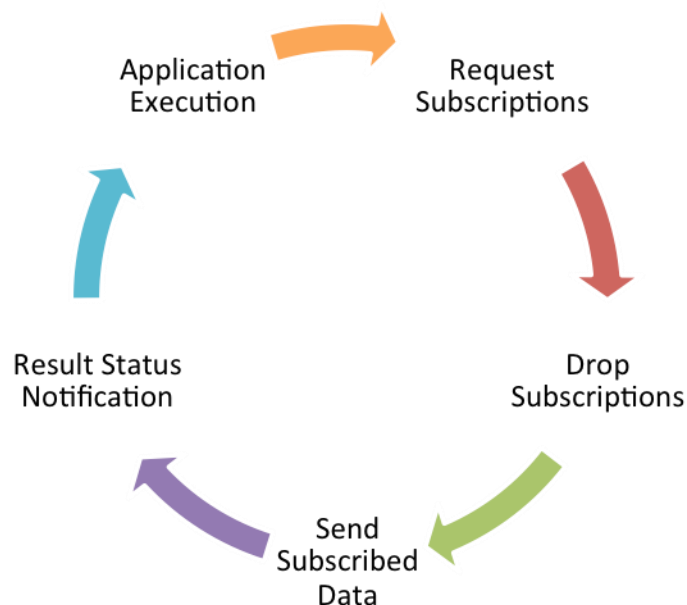


Figura 3.5: Ciclo di esecuzione di una applicazione ripetuto per ogni step di simulazione.

L'esecuzione delle applicazioni avviene ad ogni passo di simulazione: iCS itera la correzione di nodi che stanno partecipando alla simulazione nello step attuale e per ognuno di questi viene comandata l'esecuzione di ogni applicazione. In modo simile agli step di simulazione anche l'esecuzione di un'applicazione è suddivisa in passi differenti come mostrato nella Figura 3.5. Queste fasi sono ripetute in ogni applicazione per ogni nodo. In seguito sono illustrati i vari passi.

Richiesta sottoscrizioni: iCS chiede all'applicazione se vuole creare nuove sottoscrizioni per il nodo corrente, l'applicazione può rispondere in modo negativo oppure con una nuova sottoscrizione. iCS continua a richiedere nuove sottoscrizioni fino a quando l'applicazione non risponde in modo negativo. Se viene richiesta una sottoscrizione per comandare un'azione all'interno di uno dei simulatori, ad esempio l'invio di un messaggio, iCS provvede subito a schedare il comando richiesto, che sarà però eseguito al passo di simulazione successivo.

Cancellazione sottoscrizioni: iCS itera fra tutte le sottoscrizioni che l'applicazione ha richiesto il nodo corrente. Per ognuna l'applicazione può decidere se mantenerla attiva oppure se cancellarla.

Ricezione dei dati sottoscritti: per ogni sottoscrizione attiva nell'applicazione per il nodo attuale, iCS consegna i nuovi risultati, se presenti. Ad esempio vengono consegnati i messaggi ricevuti oppure le conferme della schedulazione di un messaggio. Non tutte le sottoscrizioni prevedono la restituzione di dati.

Notifica dei risultati: questa fase è utilizzata da particolari result container per notificare l'applicazione del verificarsi di certi eventi. Un result container potrebbe, ad esempio, essere delegato alla ricezione di particolari messaggi da parte dell'applicazione: in questa fase le comunica quali messaggi ha ricevuto. Non tutti i result container restituiscono dati.

Esecuzione dell'applicazione: quest'ultima fase permette di eseguire gli algoritmi presenti nell'applicazione per il nodo attuale, dopo aver ricevuto tutti i dati che l'applicazione aveva richiesto. Alla fine dell'esecuzione l'applicazione può restituire dei risultati che iCS salverà nel result container.

Questa architettura è stata sviluppata con l'intenzione di imporre meno vincoli possibili agli sviluppatori di applicazioni, infatti l'unico requisito è quello di dover aprire una socket IP per la comunicazione. Purtroppo il comportamento della interfaccia utente offerta da iCS è molto differente rispetto al modello classico offerto da altri simulatori e questo complica lo sviluppo dei applicazioni per questa piattaforma.

3.2 Architettura generale

La realizzazione di un sistema software complesso richiede necessariamente un attento sforzo di analisi per poter individuare quali sono le caratteristiche fondamentali del progetto, le soluzioni più efficaci e meno costose per realizzarlo e le modalità con cui organizzarne l'architettura complessiva. Dal momento che i requisiti possono cambiare durante la stesura del codice oppure altri imprevisti potrebbero costringere ad effettuare delle modifiche in fasi più avanzate, è particolarmente importante strutturare l'organizzazione del sistema nella maniera più modulare possibile, dividendo chiaramente i compiti e le responsabilità dei diversi componenti ed individuando le modalità più semplici per organizzarli in un'architettura funzionale e semplice.

Il comportamento di un nodo di rete che comunica e scambia informazioni con altri suoi pari può essere sempre definito in termini di un complesso automa a stati finiti. L'insieme dei protocolli sviluppati ed esposti in questo capitolo si potrebbe quindi implementare come un unico componente monolitico che riceve in ingresso i messaggi ricevuti dagli altri nodi della rete, effettua eventuali cambiamenti di stato e

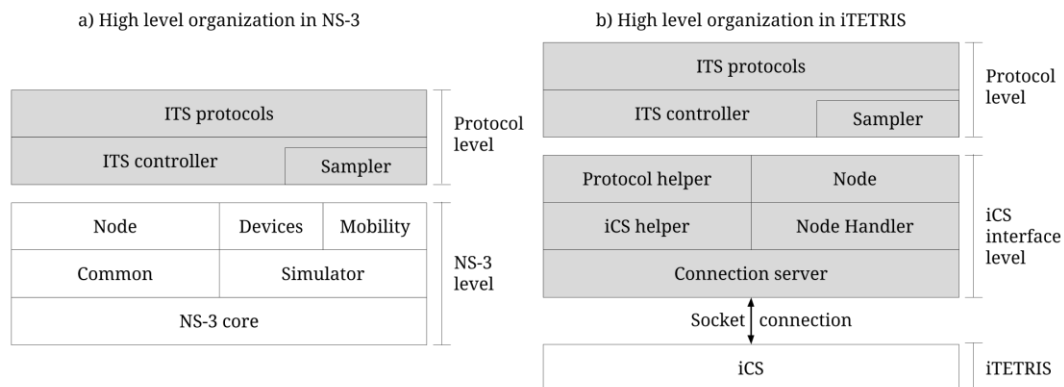


Figura 3.6: Organizzazione di alto livello dei moduli ITS sviluppati (riquadri in grigio). Organizzazione originale in NS-3 (a). Organizzazione modificata per il funzionamento in iTETRIS (b). Il livello di interfaccia con iCS permette di ridurre le modifiche da apportare al protocollo.

produce all'occorrenza dei segnali in uscita. Una simile strategia guadagnerebbe sicuramente in termini di prestazioni e risulta infatti l'approccio principalmente utilizzato quando si ha l'esigenza di sintetizzare una logica complessa in un unico componente hardware compatto ed efficiente.

Risulta tuttavia evidente come questa scelta possa rivelarsi del tutto controproducente nel momento in cui si desidera semplicemente arrivare a definire un prototipo del sistema da poter studiare e perfezionare: il punto di interesse principale di questo lavoro di tesi non è infatti l'ottenimento di un prodotto hardware o software finito e pronto per l'implementazione e l'utilizzo in larga scala ma piuttosto la realizzazione, lo studio e l'analisi di quell'insieme di protocolli di comunicazione tra veicoli introdotti nel capitolo precedente. Anche se la prospettiva di una futura sperimentazione reale è comunque tenuta in considerazione, occorre precisare che le soluzioni di seguito esposte sono state concepite in vista di una loro simulazione all'interno di uno specifico ambiente di simulazione e non sono da intendersi come una proposta di implementazione per un eventuale prototipo reale.

Il protocollo decentralizzato è stato originariamente sviluppato per essere utilizzato all'interno della versione standard di NS-3 traendo beneficio della struttura modulare di questo simulatore per realizzare un'architettura composta da molteplici componenti, ognuno incaricato dell'attuazione di un compito ben definito, come mostrato in Figura 3.6a (Riferimento tesi di Enrico Zamagni). Durante la creazione di un'applicazione iTETRIS che realizzasse le stesse funzionalità si è cercato riutilizzare l'architettura disponibile modificandola il meno possibile. A questo proposito si è deciso di utilizzare un'architettura a livelli per separare esplicitamente le funzionalità dei vari

moduli e per agevolarne lo sviluppo e la modifica. È stato quindi necessario lo sviluppo di un componente che si interponesse tra la logica dei protocolli ed iCS, come mostrato in Figura 3.6b. Questo componente offre ai protocolli un'interfaccia simile a quella offerta da NS-3 preoccupandosi di trasformare ogni comando ricevuto nel formato adatto per l'interazione con iCS.

Particolare attenzione è stata dedicata alla separazione delle funzionalità fra i due moduli dell'applicazione iTETRIS in modo da mantenerli indipendenti fra di loro consentendo la modifica e la creazione di nuovi protocolli di rete senza la necessità di intervenire sull'interfaccia con iCS (*iCS interface level*). L'implementazione del protocollo centralizzato ha infatti richiesto la sola modifica dei protocolli di rete, mantenendo inalterato il modulo di interfaccia. Ciascuno dei blocchi evidenziati in Figura 3.6 verrà in seguito ampiamente presentato ed argomentato. Per comprendere meglio l'organizzazione complessiva dell'architettura del progetto si noti come l'insieme che racchiude tutte le logiche dei protocolli sviluppati (*ITS protocols*) sia appoggiato su due componenti fondamentali che forniscono servizi indispensabili all'intero sistema. *ItsController* è il controllore del sistema sviluppato e si preoccupa di fornire una serie di funzionalità di base utili alla realizzazione dei diversi protocolli. In particolare, esso realizza le primitive per lo scambio e la ricezione dei messaggi e per il recupero delle informazioni di stato del veicolo. *NodeSampler* è un componente di fondamentale importanza per il sistema in quanto si occupa di realizzare un continuo campionamento dei valori di posizione e velocità del veicolo introducendo anche errori casuali volti a simulare realisticamente la non perfetta precisione ed accuratezza dei sensori utilizzati per recuperare tali informazioni.

Durante lo sviluppo delle logiche di comunicazione utilizzate dai protocolli (*ITS protocols*), piuttosto che produrre un unico macro componente per l'intero protocollo di gestione del ciclo di vita di un nodo, dalla formazione di un gruppo allo scambio di informazioni con il leader fino alla chiusura del gruppo stesso, si è preferito suddividere tale logica in tanti componenti più semplici ed indipendenti, ciascuno di essi incaricato di occuparsi di un preciso aspetto funzionale del sistema. Questi moduli a granularità più fine vengono definiti comportamenti e derivano tutti da un'unica classe astratta di base *Behaviour* che ne formalizza l'interfaccia e le caratteristiche comuni. È quindi previsto un comportamento esclusivamente dedicato alla formazione dei gruppi, uno che definisce la logica del nodo leader, uno che descrive il protocollo utilizzato dai nodi membri di un gruppo e così via.

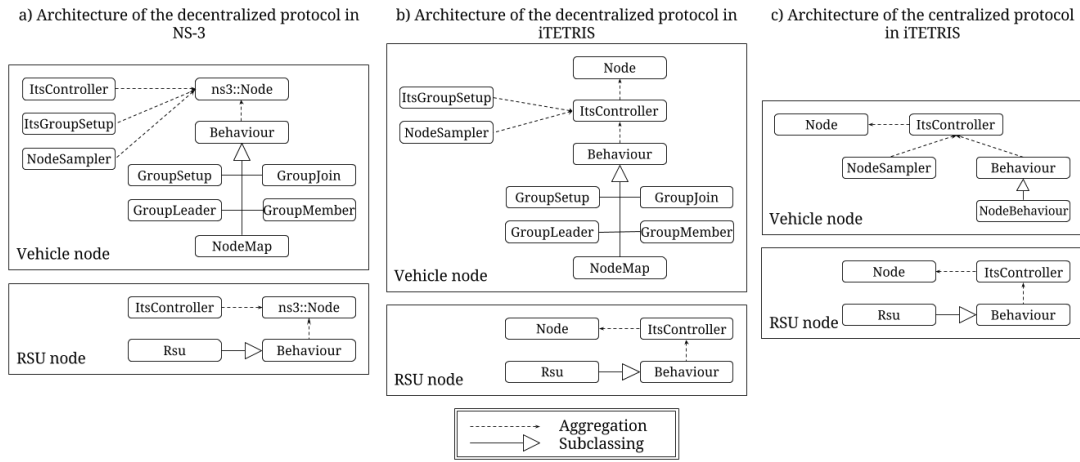


Figura 3.7: Architettura complessiva dei moduli ITS. Protocollo decentralizzato originale in NS-3 (a). Protocollo decentralizzato adattato per il funzionamento in iTETRIS (b). Protocollo centralizzato in iTETRIS (c). La modifica principale è l’aggregazione di ciascun componente alla classe `ItsController` invece che alla classe `Node`.

Tali classi di oggetti sono pensate per poter funzionare in modo completamente indipendente l’una dalle altre e per poter essere installate e attivate secondo necessità, ad esempio se si intende concentrare lo studio sulla risposta dei singoli nodi al protocollo di creazione dei gruppi è possibile attivare esclusivamente il comportamento che realizza tale logica. Questa scelta permette quindi di concentrarsi sullo sviluppo e sul test di ciascun componente in maniera modulare ed offre una notevole flessibilità ed espansibilità al sistema, potendo in qualsiasi momento abilitare e/o riconfigurare un qualsiasi set di protocolli o crearne dei nuovi partendo da zero. I diversi comportamenti sviluppati verranno descritti dettagliatamente nel capitolo 4, dopo che saranno stati chiariti gli aspetti riguardanti i componenti architetturali di più basso livello, indispensabili per il corretto funzionamento di questi moduli.

La Figura 3.7 schematizza quanto esposto finora ed espone tutti i componenti che realizzano nel complesso le soluzioni proposte. La classe `Node` costituisce il punto di partenza principale dell’architettura in quanto utilizzata per rappresentare sia le autovetture, indipendentemente dalla particolare tipologia di veicolo di cui si vuole disporre, sia le unità infrastrutturali fisse. Queste ultime, trattandosi di nodi totalmente privi di mobilità e con un insieme di funzionalità radicalmente differente rispetto ai veicoli, sono caratterizzate da una struttura dei componenti semplificata: il campionario di posizione, direzione e velocità è assente in quanto non necessario, mentre l’insieme dei comportamenti installati prevede la sola classe `BehaviourRsu` che ne

descrive autonomamente la logica. I normali nodi dispongono invece di tutti i componenti presentati in figura ad eccezione degli *shadow vehicle* che non dispongono di alcun modulo oltre al semplice oggetto *Node*: questa particolare classe vuole infatti simulare quei veicoli che non possiedono un sistema di comunicazione ITS e non possono pertanto partecipare ai protocolli di seguito descritti.

Confrontando le figure 3.7a e 3.7b si può osservare come l'architettura complessiva del protocollo decentralizzato sia rimasta praticamente immutata nonostante la transizione dall'ambiente di simulazione NS-3 a quello offerto da iTETRIS. Il cambiamento più rilevante è lo spostamento di alcune funzionalità dalla classe *Node* alla classe *ItsController* per ottenere una divisione dei compiti più chiara fra i due livelli dell'applicazione iTETRIS sviluppata. Grazie a questo cambiamento un nodo all'interno del livello di interfaccia con iCS deve semplicemente istanziare il controllore del protocollo, che si occuperà poi di gestire tutti gli altri componenti necessari al funzionamento. L'architettura del protocollo centralizzato, come si può osservare in Figura 3.7c, riutilizza molti dei concetti sviluppati per la parte decentralizzata, anche se la relativa semplicità delle interazioni realizzate ha permesso di ridurre il numero di componenti richiesti.

3.2.1 Interfaccia con iCS

L'interfaccia utente offerta da iCS e l'architettura di iTETRIS garantiscono grande flessibilità nello sviluppo di applicazioni, purtroppo però questo impone un approccio per la definizione di protocolli significativamente differente rispetto a quello che gli sviluppatori sono abituati ad utilizzare nelle versioni standard di SUMO o NS-3. Ciò rende l'utilizzo di questa piattaforma non sempre semplice e veloce, limitando l'adozione di iTETRIS nella simulazione di protocolli articolati e complessi come quelli presentati in questa soluzione.

Per facilitare l'implementazione del protocollo decentralizzato e per semplificare la futura creazione di nuovi protocolli si è deciso di sviluppare un modulo da interporre fra i protocolli di comunicazione e l'interfaccia utente offerta da iCS. L'architettura di questo componente è stata creata con l'intenzione di offrire ai protocolli simulati un'interfaccia semplificata simile a quella fornita da NS-3: in particolare una astrazione che rappresenta un nodo tramite la quale è possibile inviare e ricevere messaggi ed ottenere informazioni sul suo stato (posizione, velocità, tipo).

Questo componente realizza una serie di funzionalità in una organizzazione gerarchica come mostrato nella Figura 3.6b. Nel seguito saranno esposti in dettaglio i

componenti partenendo da quelli dedicati alla comunicazione con iTETRIS ed arrivando a quelli che si occupano dell'interazione con i protocolli sviluppati.

Server di comunicazione

Alla base dell'interfaccia sviluppata è presente una classe `Server` il cui compito principale è la gestione delle comunicazioni con iCS. All'avvio dell'applicazione questa classe crea e connette la socket IP utilizzata per l'interazione con iCS e si mette in attesa di ricevere i comandi elencati in Figura 3.5. Il formato dei messaggi inviati e ricevuti tramite la socket è costituito da una serie di byte, il server si occupa quindi di interpretarli correttamente per trasformarli secondo una rappresentazione interpretabile dal resto dell'applicazione. Ogni comando ricevuto dal server provoca una chiamata alla funzionalità corrispondente sul gestore di nodi.

Le altre funzionalità di questa classe sono quelle di incrementare correttamente lo step di simulazione, notificando gli altri componenti dell'inizio di un nuovo ciclo di simulazione, e di indicare al gestore dei nodi quando una entità è uscita dalla simulazione. iCS non indica esplicitamente quando eliminare un nodo: l'unica informazione disponibile all'applicazione è la sospensione dell'esecuzione degli step applicativi di per quel particolare nodo. Il server ogni volta che viene incrementato lo step di simulazione controlla per quali nodi l'applicazione non è stata eseguita e ne comanda l'eliminazione al gestore di nodi.

Gestore dei nodi

Questo componente, realizzato dalla classe `NodeHandler`, contiene la lista dei nodi che fanno al momento parte della simulazione. Ogni volta che il server riceve un comando, invoca la funzionalità corrispondente in questa classe, che si occupa di selezionare il nodo richiesto e di inoltrargli la richiesta. La risposta di un nodo segue il percorso inverso, infatti passa sempre per questa classe prima di essere inoltrata al server, mantenendo un livello di indirettezza fra questi due componenti.

Questa classe si occupa anche della creazione dei nodi: in modo simile a quanto avviene per l'eliminazione, iCS non informa direttamente l'applicazione dell'ingresso nella simulazione di un nuovo veicolo, occorre quindi controllare quando viene eseguito il primo step applicativo, la richiesta di nuove sottoscrizioni, per un identificativo non presente fra quelli attualmente attivi ed istanziare opportunamente il nodo. L'applicazione iTETRIS realizzata contiene ogni nodo in questo livello di interfaccia, indipendentemente dalla particolare classe di veicoli a cui appartiene, mentre nel livello di protocollo saranno esclusi i nodi di classe `shadow`, per i quali non è prevista la presenza del controllore.

Moduli di supporto ad iCS

Per configurare correttamente i componenti di questa interfaccia e le funzioni del protocollo di comunicazione implementato viene offerta una classe che si occupa di leggere il file di configurazione all'avvio dell'applicazione e di impostare gli attributi nei vari moduli che compongono l'architettura.

Come introdotto in precedenza le interazioni fra iCS e le applicazioni si basano sul meccanismo di sottoscrizioni. Ogni sottoscrizione ha un particolare formato nel quale sono specificate tutte le opzioni che è possibile controllare durante la creazione, ad esempio per registrarsi all'invio di un messaggio occorre specificare innanzitutto a che tipo di trasmissione si è interessati (unicast, geobroadcast, etc.) e successivamente indicare una serie di altri parametri a seconda della tipologia scelta. Nel caso di messaggi unicast occorre indicare la lista di nodi con i quali si vuole comunicare; nel caso di messaggi geobroadcast occorre indicare l'area di copertura sulla quale verrà propagata la comunicazione, etc. Anche le sottoscrizioni, come ogni altro messaggio, hanno un formato costituito da una serie di byte. Per semplificarne la gestione, evitare inutili ripetizioni di codice e centralizzare in un unico componente tutti i dettagli riguardanti il particolare formato utilizzato è stata sviluppata una classe statica, chiamata `SubscriptionHelper`, con lo scopo di fornire una serie di primitive, una per ogni sottoscrizione utilizzata, alle quali è possibile fornire in ingresso i parametri necessari a specificare la richiesta e ricevere in uscita un oggetto che fa da wrapper ad una collezione di byte che può essere poi inviata tramite socket dal server ad iCS.

Nodo

Il nodo è uno dei componenti più importanti di tutta l'interfaccia sviluppata, si occupa infatti di gestire le sottoscrizioni e di fornire il punto di contatto fra i due livelli dell'architettura dell'applicazione iTETRIS.

Per semplificare la creazione e l'implementazione di nuovi protocolli di comunicazione si è deciso di nascondere tutta la logica di interazione con iCS basata su sottoscrizioni e di offrire una semplice serie di primitive per svolgere i compiti di invio e ricezione di dati. La classe `Node` si occupa di trasformare le chiamate a questi metodi nelle sottoscrizioni corrispondenti, e ad informare il protocollo della ricezione dei risultati. Il nodo si occupa della gestione delle sottoscrizioni attualmente attive tramite una collezione e di richiedere la loro aggiunta e rimozione in iCS. Quando un nodo viene creato si sottoscrive automaticamente alla ricezione di messaggi geobroadcast, l'unico tipo utilizzato dai protocolli, e quando iCS consegna un nuovo risultato, il nodo si occupa di inoltrare il pacchetto al controllore del protocollo. Ogni volta che

viene richiesto l'invio di un nuovo messaggio il nodo crea la sottoscrizione corrispondente, la inserisce in una coda temporanea e la trasmetterà quando iCS richiederà le nuove sottoscrizioni per quello specifico veicolo nel successivo step applicativo. Questa coda è necessaria dato che l'interazione con il simulatore segue una sequenza ben predefinita nella quale l'applicazione non può mai prendere l'iniziativa, un nodo deve infatti attendere il proprio turno di servizio per poter comunicare con iCS. La coda è inoltre utilizzata per permettere al protocollo di inviare contemporaneamente più messaggi (ogni messaggio genera una propria sottoscrizione).

Il nodo si occupa anche di selezionare a quale classe di veicolo fa parte: nella configurazione di una simulazione si può selezionare la penetrazione percentuale desiderata per ogni tipo; durante l'esecuzione quando un nodo viene istanziato decide la propria classe attraverso una variabile casuale distribuita uniformemente fra zero e uno. A differenza delle altre classi di veicolo, se è selezionata la tipologia shadow, il nodo non si sottoscrive automaticamente alla ricezione di messaggi visto che sarebbe inutile a causa della mancanza del controllore a cui inoltrare i pacchetti ricevuti. Questa selezione automatica è valida solamente per le classi veicolo, le RSU devono infatti essere elencate nel file di configurazione assieme alle direzioni per le quali devono monitorare il flusso del traffico.

Per ottenere le informazioni sulla posizione e velocità di ogni nodo si è deciso di creare una sottoscrizione *ad hoc*, invece di utilizzare una di quelle predefinite, per semplificare ed ottimizzare questa operazione. La sottoscrizione predefinita per ottenere queste informazioni è infatti valida per un singolo step della simulazione ed occorre elencare nella richiesta per ogni nodo a quali campi si è interessati; occorrerebbe quindi creare ogni volta una nuova sottoscrizione per ognuno dei nodi attivi. Si è pertanto deciso di sviluppare una sottoscrizione permanente che restituisse tutti i dati rilevanti per ogni nodo attivo nello step corrente; occorre quindi crearla una volta sola in un singolo nodo per l'intera durata della simulazione. I dati ricevuti sono interpretati dal server e distribuiti opportunamente dal gestore dei nodi. Questa sottoscrizione è gestita dal nodo RSU perché, a differenza dei veicoli, rimane nella simulazione dall'inizio alla fine semplificandone la gestione. Questa logica di acquisizione delle informazioni sulla mobilità è completamente trasparente al protocollo che ottiene questi dati dall'istanza della classe `Node` sulla quale è installato.

L'architettura di questa interfaccia non utilizza il modello predefinito da iTE-TRIS per l'esecuzione delle applicazioni, infatti la maggior parte delle operazioni intraprese dal protocollo avvengono all'attivazione dello scheduler dopo l'incremento dello step di simulazione (il cui funzionamento sarà illustrato tra i moduli di supporto

al protocollo) e nella fase di ricezione dei dati sottoscritti, cioè quando iCS notifica al nodo quali messaggi ha ricevuto. In queste fasi il protocollo compie le azioni che aveva programmato oppure reagisce alla ricezione di un pacchetto da un altro nodo. La fase di esecuzione delle applicazioni è praticamente inutilizzata dall'interfaccia che non comunica nessun risultato ad iCS e non usa alcun tipo di result container. Si è deciso di utilizzare questo modello sia per limitare le modifiche al protocollo decentralizzato precedentemente implementato nella versione standard di NS-3, sia per fornire una logica più semplice per l'implementazione di protocolli di comunicazione.

Moduli di supporto al protocollo

Per facilitare la migrazione del protocollo da NS-3 sono state introdotte una serie di componenti di supporto che forniscono funzionalità utili per sia per i particolari protocolli sviluppati, sia per generici protocolli di comunicazione.

In NS-3 la classe `Packet` rappresenta l'astrazione di un pacchetto di dati e realizza di fatto un wrapper di un buffer di byte che descrive il contenuto informativo di un messaggio. Per evitare di dover compilare ed interpretare ogni messaggio servendosi unicamente di una mappa di byte, NS-3 mette a disposizione alcuni meccanismi di più alto livello per organizzare e recuperare informazioni all'interno dei singoli pacchetti. La classe astratta `Header` permette di leggere e scrivere un header in un pacchetto sfruttando la semplice astrazione di oggetto tipica dei linguaggi object-oriented. Per definirne un formato personalizzato è possibile estendere la classe `Header` in base alle proprie esigenze. La possibilità di poter aggiungere più oggetti header all'interno di un singolo pacchetto permette di simulare quanto avviene normalmente in uno stack protocollare, dove ciascun layer di servizio aggiunge il proprio header per descrivere il payload incapsulato. Nel sistema sviluppato per NS-3 era stato scelto di sfruttare questa funzionalità per organizzare le informazioni di un pacchetto ITS nei due principali livelli di architettura: quello delle logiche di protocollo e quello di controllore.

Per realizzare una logica simile all'interno dell'applicazione, evitando così di modificare il meccanismo di creazione dei pacchetti, si è deciso di ricreare una funzionalità simile attraverso una classe astratta `Header` ed una classe `Packet` che rappresenta una collezione di header. In questo modo il protocollo può creare ed interpretare i pacchetti con la logica esistente. Per ottimizzare la simulazione di questi pacchetti attraverso iTETRIS si è deciso di evitare l'invio dell'intero contenuto informativo al simulatore, e di utilizzare solamente un suo identificativo, per distinguere i

messaggi arrivati correttamente, e la grandezza del pacchetto necessaria alla piattaforma per simulare correttamente per quanto tempo occupare il canale durante la trasmissione dei dati. Gli identificativi sono ottenuti grazie alla classe `PayloadStorage` che contiene una collezione di pacchetti locale all'applicazione: quando un pacchetto viene aggiunto questa classe gli assegna un identificato univoco e lo inserisce nella collezione; questo id servirà poi per recuperare correttamente il pacchetto.

Quando il protocollo richiede l'invio di un messaggio fornisce il pacchetto che ha compilato al nodo che lo inserisce all'interno di questo storage ed utilizza l'id e la dimensione per popolare correttamente la sottoscrizione che sarà inviata ad iCS. Quando l'applicazione riceve un messaggio, il gestore di nodi recupera il pacchetto dallo storage tramite il suo identificativo e lo fornisce al nodo che informerà poi il protocollo. Per ottimizzare l'utilizzo delle risorse all'interno dell'applicazione esiste una sola copia del pacchetto, anche nel caso di trasmissione in geobroadcast nella quale i destinatari sono potenzialmente numerosi, occorre quindi una gestione attenta della loro vita. Nel caso di trasmissioni unicast il gestore di nodi si occupa di cancellare il pacchetto in seguito all'avvenuta ricezione dal nodo destinatario, mentre nel caso di trasmissioni che prevedono destinatari multipli è presente una procedura di pulizia all'interno del gestore dei pacchetti che viene attivata dall'incremento dello step di simulazione e che si occupa di rimuovere i messaggi che hanno superato il loro tempo di vita.

Fra le funzionalità offerte da NS-3 c'è una serie di classi che permettono di ottenere numeri casuali secondo una particolare distribuzione, in modo ripetibile fra simulazioni differenti. Il protocollo fa uso estensivo di queste classi ed è stato pertanto deciso di implementarle interamente all'interno dell'applicazione. Oltre ad evitare la modifica del protocollo è possibile confrontare i risultati ottenuti con le due differenti implementazioni del protocollo per verificarne il corretto funzionamento in iTETRIS.

Una delle funzionalità fondamentali per l'esecuzione del protocollo è la possibilità di schedulare azioni nel futuro, soprattutto per l'invio di messaggi o per permettere lo svolgimento periodico una funzione. È stato pertanto introdotta una classe `Scheduler` che permette di registrare una callback in modo che possa essere invocata in un istante futuro. Questa classe contiene una collezione di azioni registrate ordinate secondo l'istante di tempo nelle quali dovranno essere eseguite. Ogni volta che viene incrementato lo step di simulazione il server si occupa di notificare questa classe che scorrerà la propria collezione ed invocherà una alla volta tutte le azioni che devono essere eseguite. Vista l'architettura ad eventi discreti di iTETRIS il tempo di

esecuzione delle azioni schedate non è preciso, ma è suddiviso in predefiniti intervalli di tempo definiti dai successivi step di simulazione. L'ordinamento crescente della collezione permette comunque di eseguire le operazioni nella sequenza prevista.

L'ultima funzionalità di supporto per il protocollo fornisce un meccanismo che permette di realizzare un'interazione fra le classi indipendente e disaccoppiata. Tramite questo modulo un oggetto sorgente può segnalare lo svolgimento di un evento agli oggetti interessati ed comunicare alcuni dati rilevanti. Per realizzare questo modulo è stata implementata nell'applicazione un sottoinsieme delle funzionalità di tracing presenti in NS-3 [NSt14] composto da due classi principali: `TraceCallback` rappresenta una variabile contenente una lista di callback da invocare quando si vuole notificare l'avvenimento di un certo evento; `TraceManager` permette di gestire queste variabili, consentendo la registrazione ed il collegamento degli oggetti interessati alla ricezione di questi eventi. Un oggetto sorgente definisce una o più `TraceCallback` e le registra al gestore identificatole attraverso un nome; quando desidera notificare un evento invoca una di queste variabili passando i dati che è interessata a comunicare agli oggetti registrati. Un oggetto destinatario interessato ad essere notificato richiede al `TraceManager` di essere connesso con una particolare `TraceCallback`, specificando l'istanza dell'oggetto sorgente, il nome con cui è stata registrata ed il metodo che deve essere invocato dallo svolgimento dell'evento; questo è il metodo del destinatario sarà invocato e permetterà una reazione all'evento. Questa funzionalità viene utilizzata dal protocollo per collegare fra loro in modo disaccoppiato varie logiche differenti e per centralizzare la generazione dell'output della simulazione in un'unica classe.

3.2.2 Modulo di campionamento

Quasi tutte le logiche di protocollo implementate per il sistema utilizzano in maniera estensiva le informazioni di posizione, velocità e direzione introdotte nel capitolo 2 e prodotte al livello più basso dell'architettura della parte dedicata ai protocolli dell'applicazione iTETRIS dal componente `NodeSampler`. Compito di quest'ultimo è quello di effettuare una serie di campionamenti ad intervalli regolari nel tempo, introdurre degli errori casuali per simulare varie imprecisioni negli strumenti di rilevazione e di effettuare una media dei campionamenti più recenti per restituire valori meno sensibili ai disturbi ed ai cambiamenti di direzione poco marcati.

Al momento del singolo campionamento, i valori ottenuti da SUMO attraverso una sottoscrizione specifica di iCS consistono nei vettori di posizione e velocità. La

posizione corrente del nodo viene alterata sommandovi un vettore di errore caratterizzato da una direzione con distribuzione uniforme nell'intero angolo giro e da un modulo con distribuzione normale (gaussiana) a valore medio pari a zero ed un raggio massimo configurabile. L'errore sulla velocità viene invece ottenuto in due fasi: sommando al modulo un valore ricavato attraverso un errore costante alterato da una distribuzione normale con valore medio pari a zero; ruotando il vettore di velocità di un angolo distribuito normalmente e con valore medio pari a zero, scegliendo un valore di varianza molto basso per garantire che valori ottenuti dalla distribuzione si mantengano sufficientemente prossimi allo zero. In letteratura esistono dei modelli di errore estremamente più sofisticati [Ran94] che consentono di simulare in maniera molto più realistica le imprecisioni dovute ai sensori di geolocalizzazione più utilizzati. L'adozione di tali tecniche avrebbe tuttavia introdotto una complessità eccessiva ed i risultati forniti dal semplice modello appena descritto sono apparsi adeguati per gli scopi del presente lavoro di tesi. È possibile modificare l'incidenza di questi errori sui valori reali tramite opportuni parametri di configurazione che consentono di assegnare ad ogni classe¹ di veicoli una differente precisione dei sensori installati.

Terminata l'introduzione di errori, i valori di posizione e velocità vengono ulteriormente processati prima della loro memorizzazione. In questa fase, attraverso semplici calcoli trigonometrici, si procede all'estrazione dell'informazione relativa alla direzione a partire dal vettore di velocità appena ottenuto. Viene inoltre verificato che lo spostamento compiuto dal nodo tra il campionamento attuale e quello precedente sia superiore in modulo a una soglia minima prefissata. Nel caso di nodo fermo o con velocità prossima allo zero si avrebbero infatti campionamenti molto vicini nello spazio ed il loro contributo risulterebbe poco utile. Un vettore di velocità nullo non consentirebbe inoltre di calcolare la direzione corrente del nodo. Questo accorgimento consente quindi di mantenere una distanza spaziale minima tra due campionamenti successivi. Come conseguenza, negli intervalli di tempo in cui il veicolo si mantiene fermo, le informazioni relative a posizione, direzione e velocità non vengono aggiornate ed esso mantiene lo stato più recente da lui memorizzato.

Terminati tutti i calcoli e le verifiche sul campione corrente, questo viene memorizzato all'interno di un buffer circolare di dimensione configurabile. Quando il modulo è a regime, l'inserimento di un valore recente comporta l'eliminazione di quello più vecchio. Le informazioni messe a disposizione da `NodeSampler` ai livelli

¹ Esclusa la classe `shadow` che non ha questo componente.

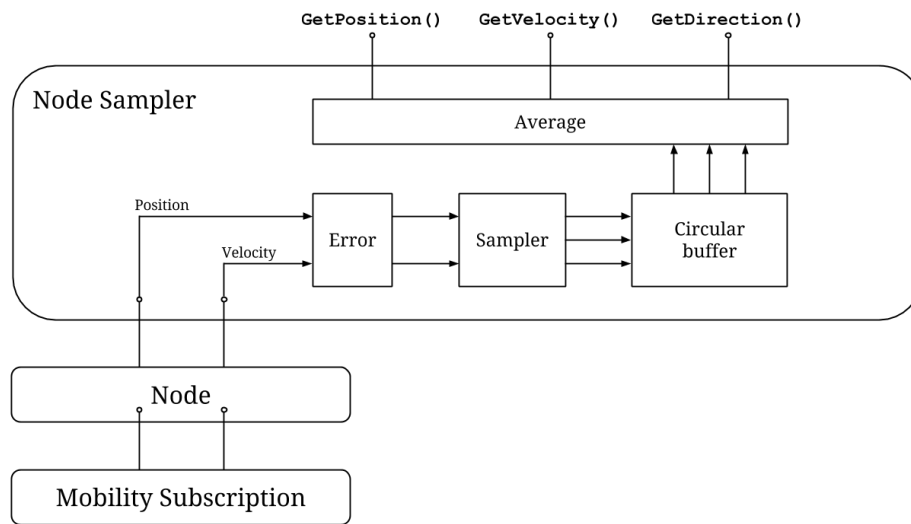


Figura 3.8: Schema di funzionamento del componente `NodeSampler`: i valori di posizione e velocità ottenuti da iCS tramite sottoscrizione sono ricevuti dal nodo corrente che li fornisce a questo componente, vengono perturbati con degli errori, viene calcolata la direzione e memorizzati all'interno dei buffer. Il valore della posizione, velocità e direzione è disponibile facendo la media dei valori memorizzati nel buffer.

superiori consistono in una media aritmetica² di tutti i valori precedentemente memorizzati. La dimensione del buffer determina quindi in maniera indiretta quanto il modulo risulta sensibile ai cambiamenti di stato sulla guida del veicolo e, soprattutto, con quale ritardo essi vengono avvertiti ai livelli più alti dell'architettura. Una memoria di grandi dimensioni si comporta inoltre come una sorta di filtro sui disturbi di breve durata in quanto questi vengono successivamente mediati tra più campioni.

Il percorso che le informazioni di posizione e velocità seguono all'interno del componente `NodeSampler` e le operazioni ad esse applicate vengono riassunte ed illustrate nella Figura 3.8 dove viene anche riportata l'interfaccia offerta alle logiche di protocollo. Occorre ricordare che i nodi di tipo RSU non sono dotati di mobilità e non prevedono pertanto il modulo per il campionamento di tali valori. Per questi nodi il controllore è consapevole dell'assenza di questo componente e provvede personalmente a fornire dei valori nulli per velocità e direzione ed a restituire direttamente la posizione, senza aggiungervi alcun errore.

² Le grandezze angolari vengono calcolate come medie circolari.

Il comportamento del modulo `NodeSampler` è progettato per offrire la massima flessibilità sui diversi aspetti delle operazioni di campionamento sopra descritte. Sia la generazione degli errori sia il comportamento del buffer circolare sono interamente parametrizzabili attraverso i molteplici attributi predisposti dalla classe ed elencati nell'Appendice A. Per caratterizzare meglio le differenti classi di veicoli introdotti nel paragrafo 2.4, con l'ovvia eccezione del tipo `shadow`, è possibile configurarle in modo differente per riflettere le differenti capacità e precisioni dei dispositivi utilizzati.

3.2.3 Controllore dei protocolli

L'interazione tra i diversi moduli raffigurati in Figura 3.6 avviene normalmente dall'alto verso il basso: le varie logiche di protocollo non comunicano direttamente tra loro essendo idealmente indipendenti ma fanno riferimento al controllore sottostante per ottenere le informazioni ed i servizi di cui hanno bisogno, mentre il `NodeSampler` rimane un componente piuttosto isolato dagli altri e si limita ad utilizzare le informazioni sulla mobilità per svolgere il proprio compito.

Nonostante sia preferibile seguire l'organizzazione verticale appena esposta, si potrebbe incorrere occasionalmente in situazioni di cross-layer dove è necessario che le logiche di più alto livello possano accedere a servizi ed informazioni presenti a diversi strati più bassi dell'architettura senza mediazione da parte di componenti intermediari. Per favorire queste eventualità ed evitare che l'interfaccia offerta dal controllore risulti troppo restrittiva e rigida, `ItsController` mette a disposizione un puntatore al nodo sul quale esso è in esecuzione che consente di accedere ad altri componenti del progetto.

Data l'importanza centrale che il controllore riveste per l'intera organizzazione delle architetture in esame, verranno di seguito elencate e discusse le funzionalità messe a disposizione ai livelli superiori.

Setup e configurazione

La configurazione di un nodo per farlo partecipare al sistema ITS sviluppato richiede la creazione del controllore e la configurazione delle singole logiche di protocollo. La configurazione dei vari moduli della architettura sviluppata viene effettuata attraverso l'utilizzo di una serie di attributi, che è possibile modificare da simulazione a simulazione senza la necessità di ricompilare l'insieme dei sorgenti. Per attributo di un modulo si intendono tutte le proprietà che si vogliono rendere configu-

rabili dall'esterno. La configurazione del sistema avviene attraverso un file di configurazione analizzato dall'applicazione iTETRIS all'avvio della simulazione per impostare il valore dei vari attributi presenti nei vari moduli del sistema. Nell'Appendice A sono elencati gli attributi per ciascun componente, ed i valori utilizzati per l'esecuzione delle simulazioni.

A differenza della versione utilizzata in NS-3, che prevede una prima fase di creazione e configurazione per ogni nodo presente nella simulazione seguita dall'effettiva simulazione dello scenario, l'applicazione iTETRIS crea e configura i nodi dinamicamente. In seguito alla creazione di un nodo vengono eseguite una serie di azioni che gli permettono di partecipare alla simulazione: viene selezionata a quale classe di veicolo appartiene attraverso la procedura descritta in precedenza e si provvede ad istanziare il controllore. Questo componente si farà carico di creare e configurare correttamente il resto dei moduli, istanziando i behaviour necessari al funzionamento della classe selezionata ed il campionatore della posizione se richiesto. Nel caso la classe selezionata sia *shadow vehicle* il controllore non viene istanziato dal nodo, che non sarà quindi in grado di prendere parte all'evoluzione dei protocolli descritti.

Attivazione e disattivazione dei nodi

Un nodo ITS è detto attivo quando è in grado di comunicare e ricevere messaggi da altri nodi attivi presenti nel proprio raggio di copertura. Nell'intervallo di tempo compreso tra l'inizio e la fine della simulazione, un nodo segue il percorso definito dalla traccia di mobilità in uso, indipendentemente dal fatto di essere attivo o meno. Il controllore offre la possibilità di attivare e disattivare un veicolo in qualsiasi momento durante la simulazione attraverso le primitive *Activate* e *Deactivate*. È inoltre possibile conoscere lo stato del nodo attraverso il metodo *IsActive*.

Grazie alla possibilità della interfaccia iCS dell'applicazione utente di creare e distruggere i nodi dinamicamente questi vengono attivati di default sia nel caso facciano parte dell'infrastruttura, che nel caso siano veicoli mobili. I veicoli verranno attivati al loro ingresso nella simulazione, in generale in un istante di tempo diverso dall'inizio della simulazione, e verranno disattivati alla loro uscita alla distruzione del nodo. I dispositivi infrastrutturali sono solitamente presenti ed attivi durante tutta la durata della simulazione.

Classi di nodi

Nel paragrafo 2.4 sono state presentate le tre classi di veicoli che sono state previste per rappresentare veicoli dotati di strumenti di comunicazione e sensori

all'avanguardia (full vehicle), utenti che utilizzano dispositivi portatili con prestazioni più contenute (medium vehicle) ed autovetture tradizionali sprovviste di ogni sistema di comunicazione (shadow vehicle). La classe `Node` introdotta in precedenza viene utilizzata per rappresentare una qualsiasi entità che partecipa ai protocolli sviluppati, sia essa un'autovettura in movimento o una RSU fissa posizionata in un incrocio stradale. Si presenta pertanto l'esigenza di garantire alle logiche di protocollo la capacità di poter distinguere la tipologia di un generico oggetto `Node` e comprendere quindi se esso è rappresentativo di una RSU o di un veicolo.

Il metodo `GetNodeType` offerto dal controllore restituisce un enumerato che descrive la tipologia del nodo sul quale esso è stato installato. I valori previsti per tale enumerato sono:

NT_RSU indica un dispositivo fisso di infrastruttura. Come mostrato nella Figura 3.7 questa tipologia è caratterizzata da un'organizzazione dei componenti differente rispetto alla controparte mobile pur disponendo della stessa interfaccia a livello di controllore.

NT_VEHICLE indica un dispositivo installato all'interno di un'autovettura. Per distinguere tra le diverse classi di veicoli previste, questo valore viene ulteriormente suddiviso come segue:

NT_FULL rappresenta la classe di veicoli full.

NT_MEDIUM rappresenta la classe di veicoli medium.

NT_ALL racchiude tutti i valori precedenti ed indica pertanto una qualsiasi entità in grado di interagire con i protocolli ITS realizzati.

NT_SHADOW indica un veicolo privo di capacità di comunicazione. Questo valore non è mai utilizzato dalle logiche dei protocolli e serve esclusivamente per indicare all'interfaccia iCS dell'applicazione iTETRIS quali nodi rappresentano veicoli shadow. Il valore **NT_ALL** introdotto precedentemente non include questa classe.

Invio di messaggi

Le logiche di protocollo installate sopra un controllore fanno affidamento alle primitive di invio e ricezione dei messaggi per realizzare lo scambio di informazioni da un nodo verso uno o più pari. La modalità più versatile e maggiormente utilizzata per produrre e spedire un messaggio consiste nell'utilizzare il metodo `Send` messo a disposizione dal controllore. È possibile specificare una tipologia di nodi destinatari

Tabella 3.1: Valori dell'enumerato `ProtocolId`.

Valore	Descrizione
<code>PID_GROUP_SETUP</code>	Formazione dei gruppi ed elezione del nodo leader.
<code>PID_GROUP_MANAGEMENT</code>	Gestione del gruppo all'interno del suo ciclo di vita.
<code>PID_HEARTBEAT</code>	Messaggi CAM tra veicoli per la creazione di una mappa di nodi dinamica durante l'uso della modalità proattiva nel protocollo decentralizzato.
<code>PID_SPEED</code>	Messaggi scambiati dai nodi nel protocollo centralizzato.

(selezionando `NT_ALL` non si impone alcun filtro), il contenuto informativo del pacchetto ed il tipo di protocollo previsto per la sua elaborazione. Il metodo `SendTo` permette invece di specificare un preciso nodo come unico destinatario del messaggio³. È prevista un'ulteriore primitiva `SendLoopback` la quale non invia un pacchetto verso la rete esterna ma lo inoltra verso altre logiche all'interno dello stesso nodo per supportarne la comunicazione diretta dove necessario. Quest'ultima primitiva non viene utilizzata all'interno del controllore del protocollo centralizzato in quanto è presente una singola logica all'interno di ogni classe di nodi.

Vista l'importanza che questo argomento riveste all'interno del sistema, le modalità con cui i messaggi vengono indirizzati, trasferiti e processati dalle singole logiche di protocollo verranno approfondite nel paragrafo seguente.

3.2.4 Scambio di messaggi

Le primitive più importanti messe a disposizione dal controllore sono senza dubbio quelle per l'invio e la ricezione dei messaggi. Queste funzionalità sono realizzate in modo molto simile all'interno dei due protocolli presentati, quando necessario le differenze saranno esplicitate nel seguente paragrafo.

Per comprenderne interamente il funzionamento è necessario chiarire le modalità con cui ogni singolo pacchetto di informazioni viene trasferito dalla particolare logica che lo ha generato (mittente) al componente adeguato per la sua ricezione (destinatario). Siccome è preferibile che i diversi comportamenti installati sopra ad un controllore restino il più possibile indipendenti tra loro, è necessario predisporre un meccanismo che consenta a ciascun mittente di specificare per ogni messaggio in

³ La trasmissione avviene in ogni caso in geobroadcast, sarà il controllore a filtrare i messaggi quando opportuno, come verrà spiegato meglio nel prossimo paragrafo.

uscita una tipologia di protocollo a cui esso si riferisce, senza che sia necessario conoscere a priori lo specifico componente designato per la ricezione. In questo modo si evita non solo di introdurre una dipendenza diretta tra le diverse logiche del protocollo ma si garantisce anche una maggiore flessibilità ed estensibilità dell'intera architettura dal momento che è possibile avere più logiche locali a uno stesso nodo come

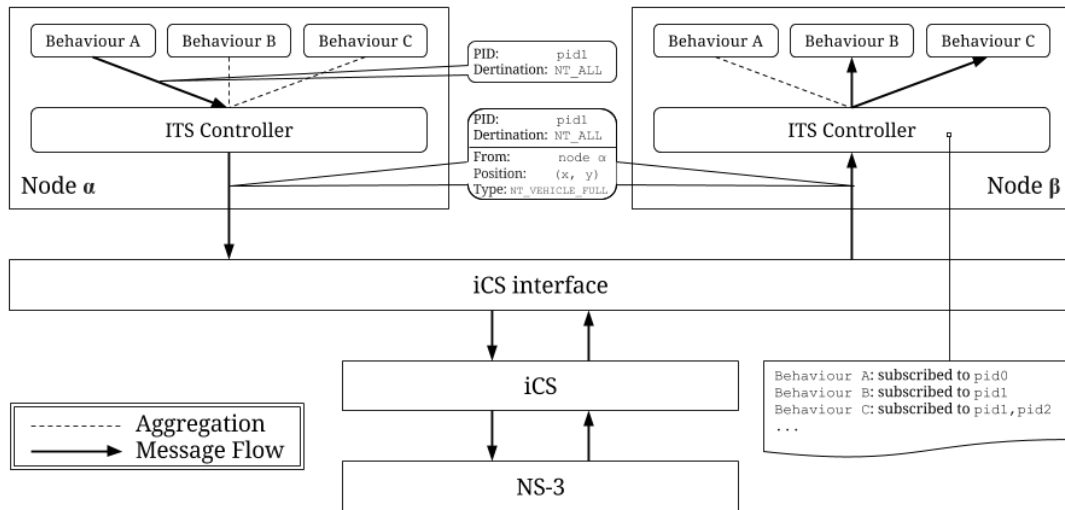


Figura 3.9: Invio ed instradamento di un messaggio ITS da un nodo α ad un nodo β.

destinatari indipendenti di un unico messaggio.

Il sistema definisce in maniera univoca diversi protocolli previsti per il sistema sviluppato attraverso il tipo di dato enumerato `ProtocolId` (`pid`) che può assumere i valori riassunti in Tabella 3.1. I primi tre valori sono utilizzati dal protocollo decentralizzato, mentre il quarto è l'unico utilizzato dal protocollo centralizzato. Per quest'ultimo protocollo si è scelto di mantenere comunque l'indicazione sulla tipologia dei messaggi, anche se non strettamente necessario, per facilitarne l'estensione ed eventualmente rendere possibile la simulazione di un sistema che utilizza contemporaneamente i due protocolli.

A ciascun messaggio inviato per mezzo del controllore viene quindi assegnato uno dei precedenti valori per indicare a che tipologia il messaggio appartiene. Al momento della ricezione di un messaggio, un qualsiasi nodo ricevente analizzerà il valore di questo campo e lo confronterà con una tabella nella quale viene indicata, per ogni behaviour iscritto al controllore, una lista di `pid` al quale tale logica è interessata. Il controllore si preoccupa quindi di consegnare il pacchetto appena ricevuto a quei comportamenti che hanno espressamente richiesto di ricevere informazioni relative a

quello specifico `pid`. A tale scopo la classe base astratta `Behaviour` prevede il metodo virtuale `Receive` attraverso il quale il controllore comunica l'avvenuta ricezione di un messaggio d'interesse. La procedura di dispatch dei pacchetti in ingresso non è tenuta a seguire un ordine preciso nella consegna alle diverse logiche in quanto indipendenti.

Questo meccanismo viene schematizzato nella Figura 3.9 dove sono raffigurati un nodo mittente α ed uno ricevente β . Il `behaviour A` all'interno del primo nodo chiede al suo controllore l'invio di un pacchetto destinato a qualsiasi tipo di entità (`NT_ALL`) relativamente al protocollo `pid1`. Il controllore di α processa tale richiesta ed invia il pacchetto sfruttando l'interfaccia ad `iCS` offerta dall'applicazione che si occuperà di effettuare tutte le azioni richieste per schedulare correttamente il messaggio per essere simulato all'interno di `iTETRIS`. Prima di consegnare il messaggio il controllore aggiunge un ulteriore header che contiene alcune informazioni utili sul mittente (maggiori informazioni nel seguito). In seguito alla ricezione del pacchetto, il controllore di β esamina la lista degli interessi dei comportamenti in suo possesso e scopre che entrambi i `behaviour B` e `C` sono interessati al protocollo `pid1` e provvede quindi ad inoltrare tale messaggio ad entrambi.

È importante sottolineare che esiste una relazione uno a molti tra un comportamento mittente ed i relativi destinatari e tra un nodo mittente ed i riceventi: un pacchetto viene inviato da uno ed un solo nodo ad opera di un solo specifico comportamento; un pacchetto viene ricevuto da zero, uno o più nodi e, localmente a ciascuno di questi, può essere processato da zero, uno o più comportamenti diversi. Questo meccanismo consente un forte disaccoppiamento tra mittente e destinatario e permette di organizzare meccanismi molto articolati (in particolare quelli di gestione dei gruppi) in più logiche indipendenti che svolgono un insieme ristretto di funzionalità complementari in sinergia con le altre.

Header

Come introdotto nel paragrafo 3.2.1, i messaggi sono scambiati dai nodi si compongono da due differenti header, uno introdotto dalla logica di protocollo, l'altro dal controllore.

A livello più alto, ogni `behaviour` aggiunge a ciascun messaggio una serie di informazioni relative allo specifico comportamento per cui è stato concepito. Tali dati vengono inviati al controllore mediante le primitive di scambio messaggi ma risultano del tutto opachi a quest'ultimo, non essendo esso in grado di interpretarli. Per ciascuna delle tipologie di messaggio descritte nella Tabella 3.1 viene realizzata un'apposita

Tabella 3.2: Campi previsti per `ItsHeader`.

Campo	Tipo	Descrizione
<code>ProtocolId</code>	enumerato	Identificativo del protocollo relativo al messaggio (Tabella 3.1).
<code>SourceId</code>	intero	Identificativo univoco del nodo mittente.
<code>DestinationId</code>	intero	Identificativo univoco del nodo destinatario (facoltativo).
<code>SourceType</code>	enumerato	Identificativo della classe di nodo mittente (paragrafo 3.2.1).
<code>Destination-Type</code>	enumerato	Identificativo del tipo di nodi destinatari (facoltativo).
<code>SourcePosition</code>	<code>Vector2D</code>	Posizione del nodo mittente al momento dell'invio del messaggio. <code>Vector2D</code> è una classe che contiene una coppia <code>double</code> che rappresentano le coordinate x ed y del nodo.
<code>SourceDirection</code>	<code>double</code>	Direzione del nodo mittente al momento dell'invio del messaggio. Assente nel protocollo centralizzato.

estensione della classe `Header` come sarà descritto nel prossimo capitolo durante la discussione dei protocolli.

A livello di controllore, ad ogni messaggio in uscita viene automaticamente aggiunto l'header `ItsHeader` che fornisce un'importante lista di informazioni relative al mittente ed al suo stato, unita ad alcune specifiche sul tipo di destinatario desiderato. Questi dati sono di norma estremamente utili anche alle singole logiche e rimangono pertanto visibili ai livelli più alti del sistema, contrariamente a quanto avviene nel modello ISO/OSI nel quale è previsto un completo isolamento tra i livelli verticali. I campi previsti per l'header ITS a livello di controllore vengono riassunti nella Tabella 3.2.

La lista delle informazioni incluse per il mittente include un valore intero identificativo univoco per il nodo sorgente, il suo tipo e la sua posizione e direzione attuali. L'id univoco per ciascun nodo viene ottenuto riutilizzando il concetto di identificativo intero previsto all'interno di iTETRIS ed assegnato automaticamente ad ogni entità che compone il sistema. Nonostante questa semplificazione, il valore è del tutto analogo a quello di indirizzo MAC per le interfacce di rete. Le informazioni sulla posi-

zione e direzione del mittente, nel caso di nodi corrispondenti a veicoli mobili, riflettono il campionamento più recente a disposizione di quest'ultimo e sono pertanto soggette ad errori e possono non essere aggiornate. Il protocollo centralizzato utilizza una versione leggermente differente di questo header nella quale manca l'indicazione della direzione attuale di un nodo in quanto si è ritenuto più opportuno spostarla all'interno dell'header di alto livello utilizzato dai behaviour.

Le informazioni sul tipo sorgente e destinatario permettono sia di identificare che classe di nodo ha inviato un messaggio, sia di inviare il messaggio attraverso un multicast in base al valore specificato in `DestinationType`. Sfruttando questo campo è quindi possibile destinare un pacchetto ai soli veicoli escludendo le RSU, oppure confinarlo ai soli veicoli di tipo full presenti nelle vicinanze e così via. È importante comunque ricordare che sono utilizzate esclusivamente comunicazioni geobroadcast, quindi i pacchetti di dati trasmessi all'interno di un canale fisico vengono sempre ricevuti da tutti i nodi iTETRIS in grado di riceverle, indipendentemente dai valori specificati all'interno dell'header. Le funzionalità di multicast e filtraggio di messaggi descritte di seguito vengono quindi realizzate esclusivamente a livello di controllore.

Il campo `DestinationId` consente infine di realizzare una trasmissione di tipo unicast verso uno specifico nodo. Messaggi di questo tipo vengono inviati mediante la primitiva `SendTo`; è possibile specificare il valore speciale `ID_ALL` per indicare che il messaggio non è esclusivo di un particolare nodo e può essere inviato come broadcast o multicast. In nessun caso vi è ovviamente alcuna garanzia di avvenuta ricezione da parte del nodo o della classe di nodi indicati come destinatari.

Filtraggio dei messaggi

L'header appena descritto viene incluso autonomamente dal controllore mittente per ogni messaggio scambiato e processato dal ricevente al momento della ricezione. Le informazioni in esso contenute vengono consegnate anche alle logiche superiori per poterle informare sull'identità e sulle caratteristiche del mittente, utili allo svolgimento dei diversi protocolli realizzati.

Prima di consegnare un messaggio agli opportuni behaviour registrati, il controllore svolge una serie di test per verificare se il pacchetto appena ricevuto è effettivamente destinato a quel nodo. Questi test consistono in un controllo sull'identificativo del destinatario, se specificato dalla sorgente, ed in un filtro relativo al tipo di

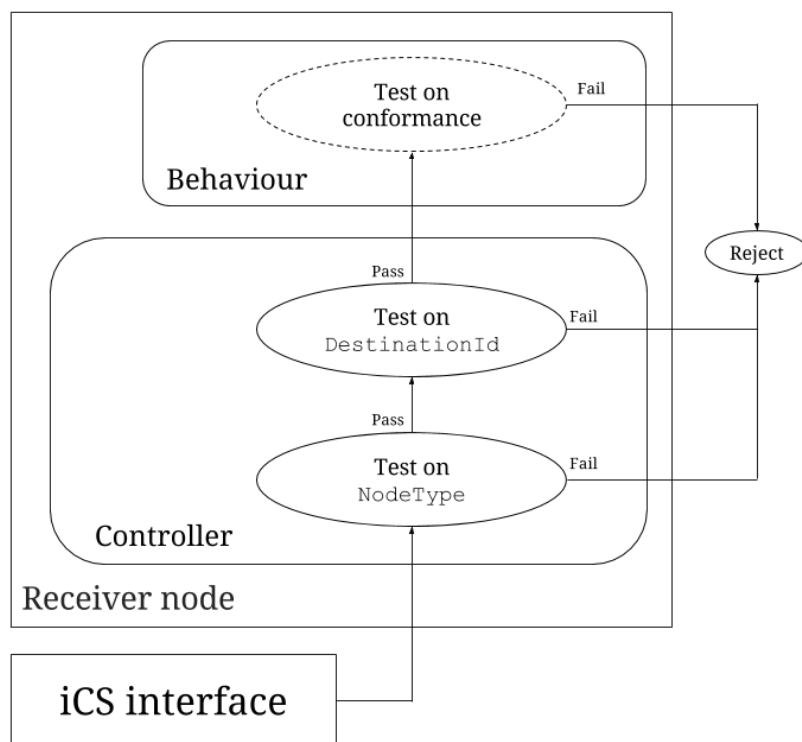


Figura 3.10: Test eseguiti su un pacchetto in ricezione. I due test effettuati dal controllore verificano se è stato specificato un mittente (*DestinationId*) o una classe di nodo (*NodeType*) come filtri di destinazione. Un singolo behaviour può effettuare un eventuale test di conformità sulla direzione.

nodo. Se entrambi i test vengono superati il messaggio è considerato valido ed è inoltrato alle opportune logiche, in caso contrario esso viene scartato. Questi passaggi vengono riassunti nella Figura 3.10.

Nel protocollo decentralizzato è inoltre possibile un'ulteriore tipologia di filtraggio sulla conformità dei valori di direzione del nodo mittente e destinatario. Questo test può essere richiesto da una logica di protocollo, la quale decide autonomamente come comportarsi in caso di successo o fallimento della verifica. Per facilitare questo controllo il controllore mette a disposizione la primitiva *IsConformantDirection* che accetta in ingresso un valore di direzione sul quale effettuare il controllo (in questo caso viene passato il valore *SourceDirection* letto nell'header) e restituisce un booleano contenente l'esito del test. Come è stato definito nel paragrafo 2.1, due direzioni sono conformanti solo se l'angolo compreso tra esse non supera il valore di tolleranza massima specificato durante la configurazione del controllore. Sfruttando questo meccanismo è quindi possibile effettuare una nuova tipologia

di messaggi multicast i cui destinatari risultano essere soltanto quei nodi che condividono la stessa direzione di marcia del nodo mittente.

Avendo presentato l'architettura complessiva del sistema realizzato, i suoi principali componenti e le meccaniche adottate per la comunicazione tra i suoi partecipanti è ora possibile procedere alla descrizione dettagliata delle logiche poste al livello superiore: i protocolli che realizzano concretamente il comportamento atteso del sistema.

Capitolo 4

Implementazione dei protocolli ITS

Nel precedente capitolo è stata presentata l'architettura generale del sistema sviluppato e ci si è soffermati sui suoi componenti principali. Lo scopo di tali moduli è quello di fornire servizi essenziali alle logiche dei protocolli poste immediatamente sopra di essi, fornendo loro informazioni di stato sul veicolo e primitive essenziali per lo scambio dei messaggi e la gestione del nodo nel quale si trovano ad operare. Verranno in questo capitolo esaminati i vari comportamenti sviluppati per il sistema, ovvero quelle logiche idealmente indipendenti tra loro introdotte nel paragrafo 3.2 le quali realizzano concretamente quei protocolli di ITS che sono stati presentati nel corso del capitolo 2. Il capitolo si sofferma prima sulle logiche sviluppate per il protocollo decentralizzato per passare poi a quelle alla base del protocollo centralizzato.

4.1 Protocollo decentralizzato

Nel seguito saranno esposte le logiche che realizzano i due comportamenti principali di questo protocollo: la formazione dei gruppi e la gestione del loro ciclo di vita.

4.1.1 Protocolli di formazione dei gruppi

Per formazione di un gruppo si intende quell'attività finalizzata alla creazione di un nuovo plotone ad opera dei veicoli che transitano in una precisa direzione, individuando quali nodi risultano compatibili con la definizione di gruppo data nel paragrafo 2.2.2 e quale tra loro viene eletto al ruolo di leader. Superata questa fase preliminare, il plotone inizierà la raccolta dei dati di interesse sulla composizione del traffico e proseguirà nel suo ciclo di vita seguendo le logiche illustrate nel paragrafo 4.1.2.

In generale, i nodi che partecipano alla formazione di un gruppo non sono isolati e nel corso delle loro comunicazioni possono ricevere messaggi provenienti da nodi già membri di altri gruppi oppure da veicoli che stanno intraprendendo la formazione di un diverso plotone. Nel primo caso, la completa separazione tra i messaggi è ga-

rantita dal campo `ProtocolId` presente nell'header principale ed indicatore del particolare protocollo a cui il pacchetto si riferisce. La formazione dei gruppi e la successiva gestione del loro ciclo di vita si appoggiano su protocolli diversi e pertanto qualsiasi interferenza tra i relativi messaggi viene evitata dal filtraggio dei messaggi illustrato nel paragrafo 3.2.4. È tuttavia necessario prevedere un sistema in grado di evitare sovrapposizioni tra due o più insiemi di veicoli impegnati in uno stesso intervallo di tempo nella formazione di gruppi separati, visto che tali nodi comunicano attraverso messaggi relativi allo stesso protocollo (`PID_GROUP_SETUP`). Tale meccanismo è inoltre estremamente utile per identificare i diversi gruppi attivi nel corso della simulazione, permette alle RSU di distinguere ed organizzare i diversi contributi ottenuti da essi e si pone come base per qualsiasi estensione al sistema che intenda ricorrere alla comunicazione tra gruppi adiacenti per realizzare funzionalità aggiuntive, alcune delle quali suggerite nel paragrafo 6.3.

Per soddisfare tale esigenza occorre notare che ogni gruppo è controllato da un unico leader ed è creato su richiesta di una specifica unità infrastrutturale alla quale vengono forniti i dati raccolti dai vari membri. È pertanto ragionevole predisporre come identificatore univoco di un gruppo un numero intero che individui sia la RSU a cui esso fa riferimento sia il nodo eletto come leader. Tale intero viene denominato `GroupId` (nel seguito abbreviato in `gid`) ed è composto dalla concatenazione dei due `NodeId` relativi rispettivamente alla RSU di riferimento ed al nodo responsabile del plotone. Questa scelta risolve a priori qualsiasi problema di duplicazione dei nodi leader dato che, per come è stato definito l'identificatore, un nodo può essere gestore di un solo gruppo ed uno stesso gruppo non può essere guidato da più di un nodo.

Durante la procedura di formazione dei gruppi, quando cioè non sono ancora stati individuati dei gestori definitivi per i nodi appena attivati, i protocolli che si occupano di questa procedura non possono ovviamente utilizzare un `gid` per coordinare la loro attività. Questo tuttavia non costituisce un problema in quanto è sufficiente utilizzare le sole informazioni relative alla direzione di riferimento ed alla RSU che ha sollevato la richiesta di creazione. Tutti i messaggi utilizzati dalle logiche di formazione includono tale coppia di valori all'interno dei loro header. Un unico messaggio beacon emesso dalla RSU può dare luogo alla creazione di più gruppi relativi ad una stessa direzione: i protocolli sono infatti progettati per indurre i veicoli a partizionarsi nell'eventualità che la loro disposizione complessiva risulti impossibile da contenere all'interno di un unico plotone. Al termine della procedura, ogni partizione assumerà l'identità di gruppo e disporrà di un `gid` univoco.

Le logiche di formazione dei gruppi si basano su due semplici principi: per evitare un eccessivo consumo di risorse energetiche e computazionali, specialmente nel caso di veicoli che sfruttano dispositivi con autonomia limitata, i nodi dovrebbero comunicare soltanto negli istanti di tempo in cui si trovano sufficientemente vicini alle unità infrastrutturali; questa osservazione è anche motivata dal fatto che i dati raccolti in posizioni eccessivamente distanti dai punti di interesse risulterebbero molto probabilmente di scarso interesse. In secondo luogo, per favorire l'identificazione del nodo leader di un gruppo, si stabilisce che il veicolo ideale per tale ruolo è quello che riesce a comunicare con il maggior numero di suoi pari, limitatamente al suo range di comunicazione ed ai nodi che possiedono una direzione conformante (cioè se l'angolo compreso tra di esse non supera un valore massimo di tolleranza, come definito nel paragrafo 2.1) a quella prevista dalla RSU. Adottando questa strategia il ruolo leader viene scelto fra i veicoli vicini al centro del gruppo stesso e si riduce il numero di plotoni che devono essere formati per coprire tutti i nodi presenti in una data direzione.

Nel paragrafo 2.2.2 sono state anticipate due diverse modalità messe a punto per la formazione dei gruppi: una reattiva ed una proattiva. La prima segue fedelmente i due principi appena esposti ed opera ipotizzando una conoscenza nulla sull'ambiente circostante da parte dei nodi inattivi che vengono attivati a seguito della ricezione di un beacon da una RSU. La seconda rilassa il primo principio per ammettere una semplice forma di comunicazione tra nodi inattivi finalizzata alla creazione di una mappa di nodi distribuita per facilitare ed accelerare la procedura di elezione del leader. Ciascuno di questi due protocolli verrà esposto nei successivi paragrafi.

Il primo passo per la formazione di un gruppo consiste quindi nell'attivare i nodi che si avvicinano ad una qualsiasi RSU in base alla direzione che questi stanno seguendo. Ciò viene realizzato dalle RSU stesse mediante un costante invio di segnali beacon ad intervalli regolari ed equidistanti. Ricordando che ogni RSU è solitamente incaricata di monitorare un insieme di più vie che si incontrano in un punto, sarà necessario che ciascuno di questi messaggi contenga una lista di valori di direzione sulla base dei quali avviare la formazione dei gruppi. Le informazioni contenute nei messaggi di beacon sono mostrate nella Tabella 4.1.

Ogni nodo inattivo che riceve un beacon verifica se la sua direzione corrente è conforme ad uno dei valori in esso indicati; se nessuno dei valori risulta compatibile il messaggio è semplicemente ignorato, in caso contrario il nodo si attiva ed inizia la successiva fase di creazione del gruppo. Un nodo inattivo può essere attivato indirettamente da un veicolo già attivo a lui conforme. Entrambe le strategie prevedono mo-

Tabella 4.1: Informazioni contenute nei messaggi beacon inviati dalle RSU.

Campo	Tipo	Descrizione
RSUId	intero	Identificatore della RSU che richiede la formazione del gruppo.
Directions	lista di double	Lista dei valori di direzione alle quali la RSU è interessata.

dalità differenti di attivazione indiretta che verranno approfondite di seguito. Il modulo che realizza una logica di tipo reattivo è denominato `BehaviourGroupSetupReactive`, mentre quello di tipo proattivo viene definito `BehaviourGroupSetupProactive`; entrambi derivano dalla classe `Behaviour` che formalizza un'interfaccia comune a tutti i moduli di protocollo e realizza alcuni servizi comuni.

Per scongiurare l'eventualità che un nodo possa fare parte di più plotoni contemporaneamente, occorre disattivare il modulo corrispondente al termine della procedura di creazione del gruppo. In questo modo i messaggi scambiati dal processo di formazione intrapreso da altri nodi sono ignorati dai veicoli già membri di un plotone in quanto il componente incaricato a processarli non è più attivo. Questa caratteristica può limitare la qualità dei gruppi formati, dato che i nodi membri non possono cambiare plotone, ma è stato deciso di accettare questa possibilità per semplificare le varie logiche realizzate. Il componente che si occupa di gestire l'attivazione e la disattivazione dei `behaviour` dedicati alla formazione dei gruppi è il `ItsGroupController`. Questo modulo collabora con il controllore principale del protocollo decentralizzato per gestire non solo i componenti dedicati alla creazione dei plotoni, ma anche quelli che si occupano del suo ciclo di vita.

4.1.1.1 Strategia reattiva

Come è stato introdotto per la selezione di un leader per un gruppo si devono preferire i veicoli che racchiudono il più alto numero di nodi all'interno del loro raggio di comunicazione. Tuttavia, al momento della loro attivazione in seguito alla ricezione di un beacon proveniente da una RSU, non è disponibile alcuna conoscenza di quali e quanti nodi risultano presenti nel range di un veicolo, né tantomeno esiste la possibilità di sapere quali tra questi possiedono una direzione conforme a quelle indicate dalla RSU.

Per ottenere questa conoscenza nel momento in cui un qualsiasi nodo viene attivato in maniera diretta, esso provvede a rispondere con un messaggio geobroadcast

contenente la propria identità e la propria intenzione a formare un gruppo nella direzione specificata. Per un certo periodo di tempo tale nodo rimarrà in ascolto di eventuali altre risposte relative alla stessa combinazione di RSU e direzione: in caso di ricezione egli è informato della presenza di un altro nodo nel suo raggio operativo che intende partecipare alla formazione di un gruppo verso lo stesso senso di marcia. Tenendo traccia del numero di risposte pervenute mediante un semplice contatore si ottiene una stima dei veicoli che tale nodo sarebbe in grado di controllare se venisse eletto al ruolo di leader.

Terminato il periodo di tempo dedicato all'ascolto delle risposte, i nodi attivi si contendono la posizione di leader mediante una sorta di asta nella quale un nodo può proporsi come leader inviando un messaggio geobroadcast di offerta che contiene il numero di veicoli percepiti. Il nodo che avrà ottenuto il valore più alto (o uno qualsiasi tra questi in caso di parità) concluderà il protocollo di elezione inviando un messaggio che sancisce la sua elezione al ruolo di coordinatore del gruppo.

Il protocollo di formazione dei gruppi nella modalità reattiva può essere suddiviso in due fasi: in una prima fase i nodi sono attivati e raccolgono informazioni su quali e quanti veicoli conformanti sono presenti nelle vicinanze; nella seconda fase procedono a stabilire quale nodo è in grado di gestire il maggior numero di veicoli tra quelli appena attivati e lo eleggono come leader. La selezione del coordinatore del gruppo conclude il protocollo di formazione.

Per illustrare la prima delle due fasi sopra introdotte si farà riferimento alla Figura 4.1 dove sono raffigurati alcuni nodi inizialmente inattivi che si avvicinano a una RSU seguendo una delle tre direzioni di interesse. Nella prima parte della figura, la RSU invia un messaggio beacon che viene ricevuto dai nodi *a* ed *f*. Quest'ultimo ha una direzione che non è conforme a nessuna tra quelle specificate nella lista e pertanto ignora il messaggio. Il nodo *a* è invece conforme alla direzione -160° e viene quindi attivato in maniera diretta. Nella Figura 4.1b, il nodo reagisce all'attivazione mediante l'invio di un messaggio di risposta che indica la propria disponibilità a formare un gruppo per la combinazione RSU e direzione specificata. Le informazioni contenute nei messaggi di risposta sono mostrate nella Tabella 4.2.

Alla ricezione di un messaggio di risposta, un nodo inattivo può attivarsi in maniera indiretta a patto che la propria direzione sia conforme a quella indicata nel campo *Direction*. Se il test di conformità ha successo ed il nodo viene attivato si dice che la richiesta di formazione del gruppo ha subito un salto di attivazione (hop), ovvero

Tabella 4.2: Informazioni contenute nei messaggi beacon inviati dalle RSU.

Campo	Tipo	Descrizione
RSUId	intero	Identificatore della RSU che richiede la formazione del gruppo.
Direction	double	Direzione per la quale si è interessati alla formazione del gruppo.
Hops	intero	Numero di salti di attivazione a partire dalla risposta originale.

tale veicolo è stato attivato non da un beacon proveniente dalla RSU, ma da un messaggio di risposta di un suo pari. In simili casi, il contatore che si occupa di tenere traccia dei nodi rilevati viene inizializzato ad uno.

Per evitare che il sistema di attivazione indiretta possa procedere in cascata, provocando l'attivazione di nodi anche molto lontani dalla RSU (flooding), si impone un limite massimo al numero di salti di attivazione possibili per una richiesta, sfruttando un concetto simile a quello di TTL [Tan02] previsto per il protocollo IP: ogni messaggio di risposta indica in un campo denominato `Hops` il numero di salti di attivazione che sono stati effettuati fino a questo momento. Se un nodo inattivo riceve una risposta con un valore di hop superiore al limite massimo prestabilito, il messaggio viene scartato e la catena di attivazioni indirette viene interrotta. Nella figura (parti b, c, d) si può notare come il nodo `a` imposta tale valore a zero nel suo messaggio di risposta in quanto attivato direttamente dalla RSU. I nodi `b` e `c` vengono attivati indirettamente ed impostano il valore ad uno e così via per i restanti nodi `d` ed `e`.

Con una semplice variazione del valore massimo ammesso per i salti di attivazione è possibile alterare il comportamento del sistema per fare in modo che una richiesta di attivazione arrivi fino a nodi molto distanti oppure imporre che questa resti confinata su un'area più limitata. In ogni caso un salto di attivazione può avvenire soltanto nel caso in cui i veicoli che ricevono il messaggio risultano avere una direzione conformante a quella stabilita inizialmente dai nodi attivati per via diretta dalla RSU. Negli scenari d'uso reale, a meno di rettilinei particolarmente lunghi, la conformazione del reticolo stradale non prevede lunghi tratti a direzione uniforme a partire dagli incroci semaforici; ne consegue che è la particolare topologia dell'ambiente che circoscrive in primo luogo la capacità di una richiesta di formazione di un gruppo di

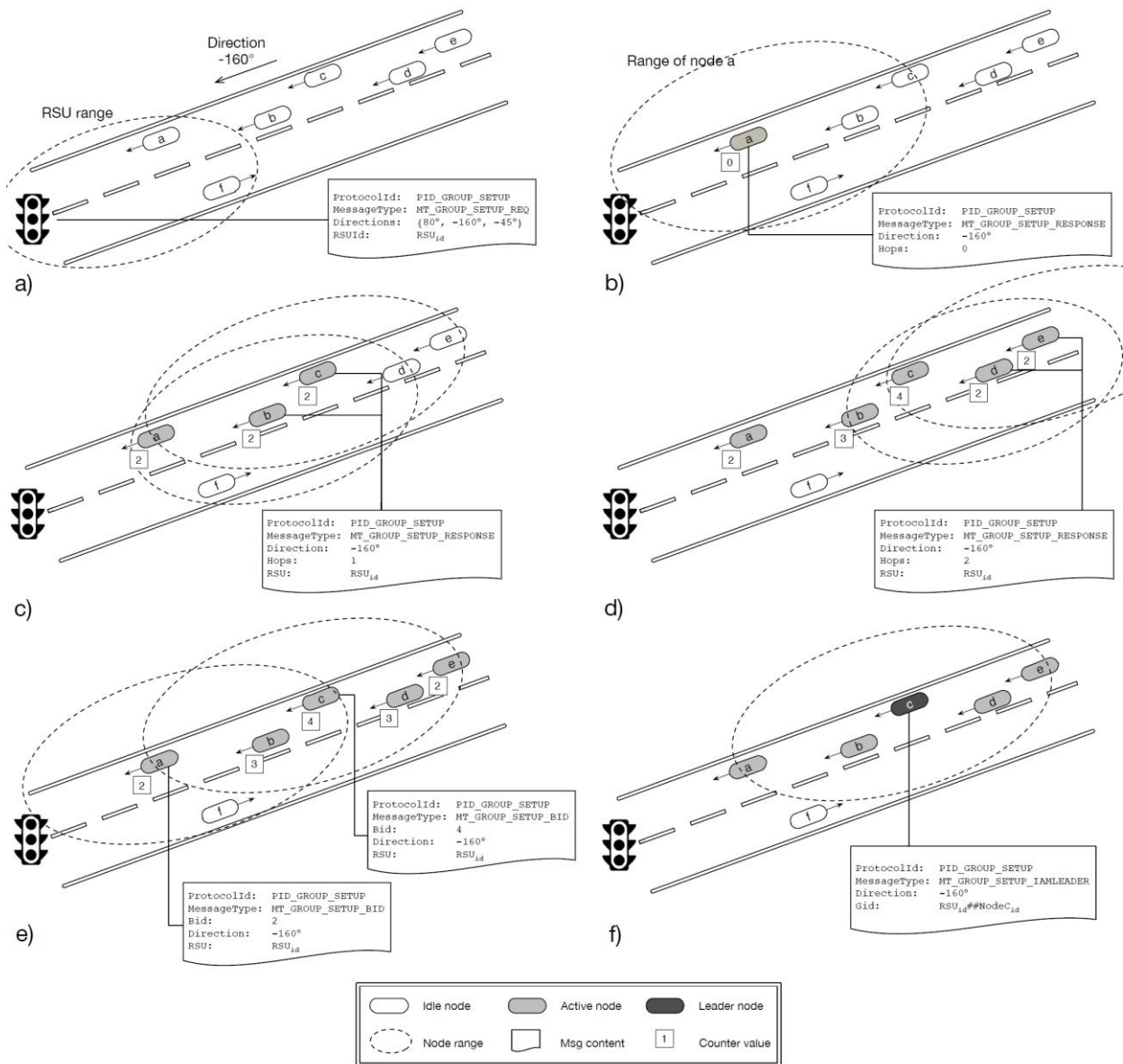


Figura 4.1: Schematizzazione del processo di formazione dei gruppi con la strategia reattiva. La RSU invia beacon periodici che attivano i nodi nelle direzioni specificate (a). Ogni nodo attivato invia un messaggio di risposta per informare i propri vicini della propria partecipazione al protocollo (b). Eventuali nodi inattivi che ricevono tale messaggio sono attivati indirettamente e rispondono a loro volta purché non venga raggiunto il limite massimo di due hop (c, d). Segue una fase di contesa del ruolo di leader mediante un'asta sul numero dei nodi percepiti (e). Il nodo che ha inviato l'offerta migliore viene eletto come leader del gruppo appena formato (f).

allontanarsi dalla RSU che la origina, indipendentemente dal limite che viene manualmente imposto ai salti di attivazione.

Tabella 4.3: Informazioni contenute nei messaggi di offerta.

Campo	Tipo	Descrizione
RSUId	intero	Identificatore della RSU che richiede la formazione del gruppo.
Direction	double	Direzione per la quale si è interessati alla formazione del gruppo.
Bid	intero	Numero di nodi che il mittente ha contato durante la prima fase.

Un ulteriore parametro che influenza il comportamento del protocollo durante questa prima fase è la durata dell'intervallo temporale nel quale i nodi raccolgono passivamente le richieste provenienti da loro pari. Superato questo intervallo di tempo la prima fase viene completata ed i nodi ritornano a comunicare per stabilire chi tra loro verrà eletto come coordinatore del gruppo. L'evento che segna il passaggio dalla prima alla seconda fase coincide con l'invio del primo messaggio di offerta, ovvero quando uno tra i nodi attivati termina l'attesa di ulteriori messaggi di richiesta e segnala la propria candidatura indicando quanti veicoli esso è riuscito ad individuare.

Per evitare che i nodi più vicini alla RSU siano avvantaggiati rispetto a quelli più distanti, si stabilisce che la durata di tale intervallo temporale sia scelta casualmente da ogni nodo mediante una distribuzione uniforme tra un valore minimo e massimo. Questa scelta aiuta inoltre a scongiurare che più nodi si sovrappongano tra loro ed inviino entrambi un messaggio in istanti di tempo talmente ravvicinati da provocare una collisione tra le loro trasmissioni. Per lo stesso motivo è prevista l'introduzione di un piccolo ritardo casuale, ottenuto mediante una distribuzione uniforme fra zero ed un valore massimo, introdotto dal nodo prima di rispondere ad una richiesta di formazione di un gruppo sia nel caso di attivazione diretta da una RSU che indiretta da parte di un altro veicolo. L'utilizzo di intervalli e ritardi casuali ricorre numerose volte nelle varie logiche di protocollo; esso permette di attenuare i problemi derivanti dalle collisioni tra pacchetti inviati in istanti troppo ravvicinati.

Il primo nodo che invia un messaggio di offerta inizia la seconda fase e risveglia tutti i veicoli nel suo range di copertura comunicandogli il numero di nodi che ha rilevato durante la prima fase. A questo punto si avvia una sorta di asta nella quale un nodo, se ha contato un numero di veicoli maggiore rispetto a quelli indicati nei messaggi di offerta precedentemente ricevuti, può rilanciare inviando il proprio valore. Per fare in modo che i vari contatori rimangano stabili durante questa procedura, ogni

Tabella 4.4: Informazioni contenute nei messaggi di elezione del leader.

Campo	Tipo	Descrizione
Group	Group	Struttura dati che contiene l'identificativo e la direzione del gruppo.

nodo che passa dalla prima alla seconda fase diventa insensibile ad ulteriori messaggi di risposta e mantiene pertanto inalterato il proprio conteggio. Quando il veicolo che dispone del contatore più alto avrà inviato la propria offerta nessun altro nodo nelle sue vicinanze sarà in grado di rilanciare; esaurito un breve periodo di timeout, l'autore dell'offerta più alta dichiarerà la propria vittoria con un messaggio geobroadcast ed assumerà il ruolo di leader del gruppo, terminando con successo il protocollo di formazione.

In maniera analoga a quanto già descritto per la prima fase, immediatamente dopo l'invio di un messaggio di offerta il mittente innesca un timer di durata casuale scelta uniformemente tra un valore minimo e massimo. Terminato questo periodo, se l'offerta precedentemente trasmessa non è stata superata, esso assume il controllo del gruppo ed invia un messaggio conclusivo per informare i nodi vicini della sua elezione e per identificare il gruppo appena formato. I nodi che ricevono tale pacchetto hanno quindi conferma definitiva dell'identità della RSU alla quale faranno riferimento, dell'id del leader per il gruppo e del valore di direzione che occorre mantenere. Le informazioni scambiate nei messaggi di offerta e di elezione del leader sono mostrate nelle tabelle 4.3 e 4.4.

Durante il periodo di attesa che precede l'invio del messaggio conclusivo di leadership, può accadere che un nodo che ha già presentato la propria candidatura, o che è in procinto di inviarla, riceva un messaggio di offerta contenente un valore superiore al proprio contatore. In entrambi i casi ogni nodo che vede la propria offerta superata da un altro veicolo deve annullare il timer che precede l'invio del messaggio di leadership ed attendere in maniera passiva che un altro nodo completi il protocollo dichiarandosi coordinatore del gruppo. Normalmente un nodo che non ha ancora inviato un'offerta recede dall'asta anche se riceve un messaggio con valore pari al proprio contatore; in ogni caso, nella eventualità che si verifichino delle parità tra due offerte, si stabilisce per convenzione che il vincitore è il nodo con l'identificatore più alto tra i contendenti.

È chiaramente preferibile evitare che la seconda fase del protocollo di elezione degeneri in un flooding di messaggi offerta. Si vuole cioè impedire l'invio eccessivo

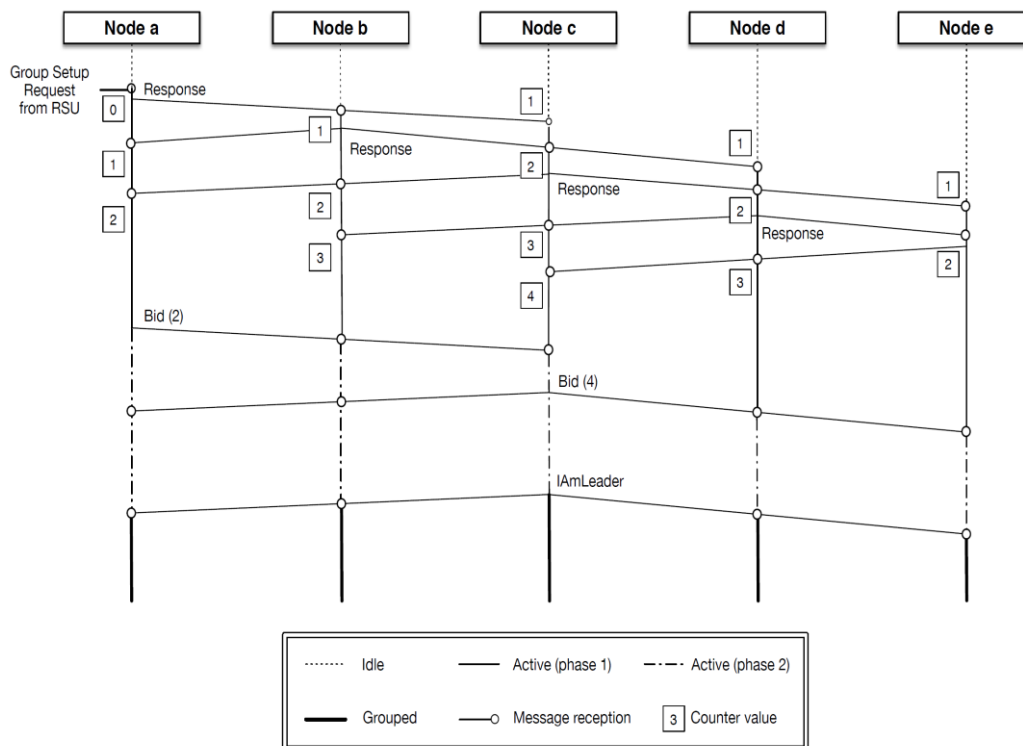


Figura 4.2: Diagramma della sequenza dei messaggi scambiati tra i nodi di Figura 4.1 durante la formazione del gruppo.

ed incontrollato di candidature che porterebbe la maggior parte dei nodi attivi a comunicare agli altri il valore del proprio contatore interno. Per regolamentare ed ottimizzare questa fase è stato stabilito che ciascun nodo attivo ha a disposizione un solo messaggio di offerta che può essere inviato esclusivamente in due occasioni: allo scadere del tempo di attesa che separa la prima e la seconda fase, oppure per rilanciare un'offerta più bassa di quella in proprio possesso.

Ciascun nodo, analogamente a quanto avviene nella prima fase, deve attendere un breve intervallo di tempo casuale prima di poter trasmettere la propria controfferta per limitare la possibilità di interferenza fra trasmissioni diverse. La ricezione da parte di un nodo di un'offerta pari o superiore alla propria durante questa attesa, causa la cancellazione dell'invio del messaggio.

Le regole appena definite prevedono un'unica eccezione nel caso in cui un nodo attivo non rilevi alcun veicolo nel proprio raggio operativo: in questa eventualità l'elezione del leader per il gruppo è banale e non è necessario ricorrere all'invio della propria candidatura. Se un nodo termina la prima fase e dispone di un conteggio dei veicoli pari a zero, esso attende comunque l'intervallo di tempo previsto per l'asta,

terminato il quale potrà procedere all'invio del messaggio leadership senza aver prima comunicato la propria offerta.

Nel diagramma mostrato nella Figura 4.2 sono rappresentati i messaggi scambiati dai nodi durante il processo di formazione del gruppo illustrato in Figura 4.1.

4.1.1.2 Strategia proattiva

In condizioni di densità di traffico molto elevate, il protocollo di formazione dei gruppi appena esposto presenta alcuni svantaggi, è stato quindi considerato lo sviluppo di una sua versione alternativa per cercare di ottenere migliori performance negli scenari particolarmente congestionati per numero di nodi che si avvicinano ad una stessa RSU. Le principali differenze tra le due strategie risiedono nelle diverse ipotesi dalle quali esse traggono ispirazione e nelle differenti modalità di scelta degli istanti di tempo per l'invio dei messaggi.

Lo svantaggio principale introdotto dalla strategia reattiva è l'elevato numero di messaggi di risposta che vengono inviati in un intervallo di tempo relativamente breve. Per poter misurare con quanti nodi ciascun veicolo è in grado di comunicare, ognuno di essi deve infatti inviare un pacchetto per informare i suoi vicini della propria presenza. Questo comportamento, nel caso di densità particolarmente elevate, può portare ad un elevato numero di messaggi tale da compromettere seriamente l'esito del protocollo di formazione a causa delle numerose sovrapposizioni fra le trasmissioni a livello fisico. Tali collisioni possono inoltre investire numerosi nodi impegnati in altri protocolli, diminuendo così le performance dell'intero sistema su un'area piuttosto vasta.

Per tentare di ridurre queste difficoltà, il protocollo di formazione di tipo proattivo parte dal presupposto che i nodi in avvicinamento alla RSU conoscano a priori i diversi veicoli presenti nelle loro vicinanze ed il numero di nodi visti da ciascuno di essi. Sotto tale ipotesi, l'elezione del leader per un gruppo diventa un'operazione estremamente più semplice in quanto ogni nodo, nel momento stesso in cui viene attivato, è in grado di sapere se esso è o meno il candidato ideale per tale ruolo. In caso affermativo esso può dichiararsi come leader dopo aver atteso un breve intervallo di tempo; se un nodo è invece a conoscenza di un candidato migliore spedisce a quest'ultimo un messaggio di invito per attivarlo. Alla ricezione di tale messaggio un nodo verifica personalmente di essere conforme alla direzione prevista per il gruppo ed in caso affermativo attende un certo intervallo di tempo prima di concludere la procedura di formazione del gruppo dichiarandosi leader, se non viene nel frattempo preceduto da un altro veicolo.

La possibilità di disporre di una conoscenza dei nodi presenti nelle vicinanze anche nei momenti in cui questi risultano inattivi è ottenibile soltanto se si ammette di poter consentire uno scambio di messaggi tra nodi idle, ipotesi precedentemente esclusa dal primo dei due principi esposti all'inizio del paragrafo 4.1.1. Per poter utilizzare la strategia proattiva si deve quindi poter rilassare questo vincolo e permettere ad ogni veicolo presente nella simulazione, ad ovvia eccezione di quelli di tipo shadow, l'invio, ad intervalli regolari ed opportunamente distanziati, di messaggi, denominati heartbeat, al fine di informare periodicamente i nodi vicini della propria presenza e di scambiare altre informazioni utili (direzione, posizione, nodi conosciuti, etc). In questo modo ciascun veicolo è in grado di creare e mantenere aggiornata una propria mappa dei nodi vicini e di conoscerne in ogni momento le loro caratteristiche.

Il servizio di mappatura dei nodi è fornito dal componente `BehaviourNodeMap` il cui funzionamento ed i dettagli sulle modalità con cui esso realizza le funzionalità sopra accennate verranno approfonditi nel paragrafo 4.1.1.3. Per il momento si assumerà di potere utilizzare una struttura dati, denominata `NodeMap`, che consiste in una collezione di nodi conosciuti con le loro relative proprietà, tra cui posizione, velocità e numero di veicoli da loro rilevati. Ovviamente queste informazioni costituiscono una stima necessariamente imperfetta dello stato reale dell'ambiente in quanto la loro freschezza è in proporzione diretta alla frequenza con cui vengono ricevuti i messaggi di heartbeat di ciascun nodo. Esse risultano comunque di grande aiuto per lo svolgimento del protocollo esposto di seguito.

La seconda differenza tra le due strategie di formazione dei gruppi è la modalità con la quale vengono scelti gli intervalli di attesa nelle varie fasi del processo di elezione. Nella logica di tipo reattiva tali intervalli vengono sempre scelti in maniera uniforme tra un valore minimo e massimo, indipendentemente dalle particolari condizioni dei veicoli che partecipano al protocollo. In generale questa scelta garantisce comunque una buona resa a fronte di un costo implementativo molto contenuto; esiste comunque l'eventualità che risultati particolarmente sfortunati portino a un'elezione non ottimale, ma questa è occasionalmente tollerabile se si considera che lo scopo ultimo delle diverse logiche implementate non è quello di arrivare ad una perfetta conoscenza delle condizioni del traffico dell'ambiente ma di ottenerne un'approssimazione utilizzabile da euristiche di più alto livello.

Nel momento in cui un elevato numero di veicoli si trova a dover scegliere un istante per l'invio di un messaggio attraverso la stessa procedura uniforme, si possono verificare frequenti collisioni tra le trame dei pacchetti, visto che la probabilità che

due o più nodi scelgano dei valori temporali troppo poco distanti tra loro cresce velocemente con l'aumentare dei partecipanti. Poiché il protocollo di tipo proattivo si propone come alternativa maggiormente scalabile in condizioni di elevata densità si è scelto di ricorrere ad un metodo differente per il calcolo dei periodi di attesa casuali.

L'intervallo di attesa più significativo per l'elezione proattiva è quello che precede l'invio del messaggio di leadership trasmesso da un candidato per dichiararsi coordinatore del gruppo in via definitiva. Un nodo risulta tanto più avvantaggiato nell'elezione al ruolo di leader quanto più breve è l'intervallo di tempo che deve attendere prima potersi proclamare tale. Nel calcolo del valore di questa attesa viene quindi tenuto conto del numero di nodi conformanti e non appartenenti ad un gruppo che il candidato può contattare in modo da avvantaggiare quelli con un valore più alto. Al valore così calcolato viene infine sommata una piccola quantità casuale, distribuita uniformemente e con modulo massimo configurabile, per evitare che due nodi con le stesse caratteristiche scelgano lo stesso valore sincronizzando le loro trasmissioni.

L'intera procedura di formazione proattiva si svolge in un'unica fase e risulta concettualmente più semplice se confrontata con quella di tipo reattiva, grazie soprattutto alle ipotesi di partenza che sono state assunte. L'attivazione diretta dei nodi da parte dei messaggi beacon provenienti dalle RSU avviene analogamente a quanto esposto in precedenza: ogni ricevente verifica la propria conformità rispetto ad una delle direzioni specificate ed in caso affermativo si attiva e dà inizio alla procedura di elezione. A differenza di quanto avviene nella strategia reattiva, nel protocollo proattivo i nodi non reagiscono all'attivazione con un messaggio di risposta ma effettuano una scansione della NodeMap locale alla ricerca di un candidato ottimale.

Il parametro che viene utilizzato da ciascun nodo come indicatore di preferenza per il ruolo di leader è la cosiddetta densità netta¹, ovvero il numero di veicoli non raggruppati che un nodo vede all'interno del proprio raggio tali per cui la loro direzione attuale risulti conforme a quella scelta nel processo di formazione del gruppo. Ad esempio, se un veicolo viene attivato per la formazione di un gruppo relativamente alla direzione di valore 30° esso conterà nella sua mappa locale il numero di nodi a lui conosciuti che non fanno già parte di un gruppo e che procedendo in una direzione di marcia conforme a 30°. Se a seguito di tale ricerca un nodo attivo scopre di essere a conoscenza di un veicolo con una densità netta maggiore alla propria esso

¹ Nel paragrafo 4.1.1.3 saranno definiti meglio i concetti di densità utilizzate dal sistema.

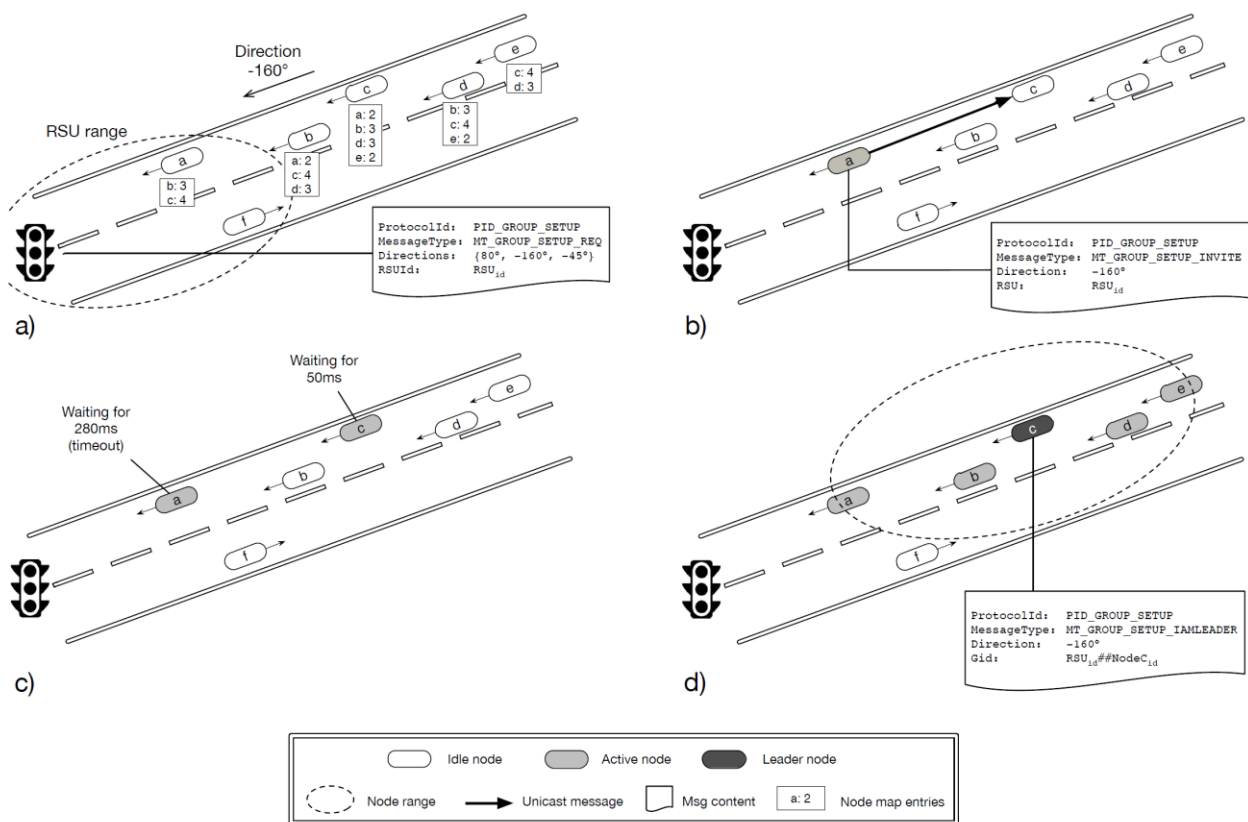


Figura 4.3: Schematizzazione del processo di formazione dei gruppi con strategia proattiva. La RSU invia beacon periodici che attivano i nodi nelle direzioni specificate (a). Ogni nodo attivato verifica la conoscenza di un valido candidato all'interno della propria mappa di nodi ed eventualmente invia un messaggio di invito esclusivamente al candidato scelto (b). Il primo nodo che termina il proprio periodo di attesa o esaurisce il timeout (c) assume il ruolo di leader inviando un messaggio geobroadcast (d).

procederà ad inviargli un messaggio unicast² di invito per comunicargli la propria preferenza. Le informazioni contenute in questi messaggi di sono mostrate nella Tabella 4.5.

Nel caso scelga di spedire un messaggio di invito, il mittente delega di fatto la creazione del gruppo al suo candidato e rimane passivamente in attesa di un messaggio di leadership che annunci il completamento della procedura, indipendentemente dall'identità del nodo che assumerà il ruolo di leader. Per scongiurare situazioni di stallo i mittenti dei messaggi di invito prevedono in ogni caso un periodo di timeout

² È utilizzata una normale trasmissione geobroadcast dall'interfaccia con iCS per inviare tale pacchetto, ma grazie alle modalità che permettono di filtrare i messaggi esposte nel paragrafo 3.2.4 solo il nodo destinatario considererà tale pacchetto. In questo senso la trasmissione si può considerare come unicast.

Tabella 4.5: Informazioni contenute nei messaggi di invito.

Campo	Tipo	Descrizione
RSUId	intero	Identificatore della RSU che richiede la formazione del gruppo.
Direction	double	Direzione per la quale si è interessati alla formazione del gruppo.

terminato il quale procederanno essi stessi a prendere il controllo del gruppo effettuando il geobroadcast del pacchetto di leadership. Tale intervallo di timeout è ottenuto attraverso un valore base configurabile sommato ad una piccola quantità casuale distribuita uniformemente e con modulo limitato ad un valore massimo selezionabile.

Alla ricezione di un messaggio di invito un nodo provvede ad attivarsi, se non lo è già, sempre verificando che la propria direzione sia conformante con quella specificata originariamente dalla RSU. Prima di poter assumere il ruolo di leader, un nodo attivato mediante invito deve attendere per un periodo di tempo scelto in maniera inversamente proporzionale al suo valore di densità netta. È infatti assai frequente che all'interno di una stessa procedura di formazione più nodi decidano di inviare molteplici messaggi di invito, mai più di uno a testa, verso candidati differenti; ciascuno di questi potrebbe decidere di assumere il comando portando ad un'inefficiente frammentazione del gruppo. Per favorire l'elezione del leader migliore si sceglie di garantire un tempo di attesa più breve ai candidati più promettenti, ovvero a quelli che conoscono un maggior numero di potenziali membri per il gruppo.

Il primo nodo che invia un messaggio di leadership assume il ruolo di leader per il gruppo ed informa tutti i nodi nelle vicinanze della sua elezione, anche eventuali veicoli ancora inattivi. I rimanenti nodi attivi che ricevono tale comunicazione annullano qualsiasi tipo di attesa o timeout e cedono il controllo del plotone al mittente. In Figura 4.3 viene schematizzata l'intera procedura proattiva di formazione di un gruppo.

4.1.1.3 Mappatura dei nodi circostanti

Nell'architettura realizzata si presenta in due occasioni la necessità di gestire e mantenere aggiornata una struttura dati locale ai singoli nodi che permetta loro di conoscere quanti e quali veicoli sono presenti in un dato momento all'interno del proprio range di trasmissione. La prima occasione, illustrata nel precedente paragrafo, si manifesta nella necessità del modulo per la formazione proattiva dei gruppi di conoscere la densità netta dei nodi conosciuti per procedere rapidamente all'elezione del leader.

Tabella 4.6: Informazioni disponibili all'interno della struttura dati `NodeInfo`.

Attributo	Tipo	Descrizione
<code>NodeId</code>	intero	Identificatore univoco del nodo.
<code>GroupId</code>	intero	Eventuale identificatore del gruppo al quale il nodo.
<code>Speed</code>	double	Velocità attuale del nodo.
<code>Direction</code>	double	Direzione attuale del nodo.
<code>Position</code>	<code>Vector2D</code>	Posizione attuale del nodo.
<code>LastSeen</code>	intero	Istante temporale nel quale sono state pervenute le informazioni più recenti per il nodo.
<code>Density</code>	<code>Density</code>	Struttura dati che contiene i valori di densità assoluta, conformante e netta relativi al nodo.

La seconda occasione si manifesta nella gestione del normale ciclo di vita di un gruppo, come sarà chiarito nel paragrafo 4.1.2. In particolare, il controllore di un plotone deve necessariamente disporre di una mappa dettagliata dei vari membri del gruppo per poter svolgere correttamente le proprie attività ed integrare i dati a sua disposizione.

Come accennato in precedenza il modulo del sistema che permette di disporre di tali informazioni è il componente `BehaviourNodeMap`, il quale si occupa di organizzarle in una struttura dati, denominata `NodeMap`, costituita da una lista di elementi di tipo `NodeInfo`, ognuno dei quali contiene i dati più recenti disponibili per un particolare nodo.

Questo modulo è piuttosto atipico se confrontato con le altre logiche di protocollo per il fatto che esso offre delle funzionalità agli altri comportamenti e non è pertanto perfettamente isolato. Esso è in grado di svolgere il proprio operato in maniera del tutto indipendente ma quei behaviour che ne fanno uso necessitano di un suo riferimento per recuperare la mappa dei nodi quando necessario.

Le informazioni di stato disponibili, elencate in Tabella 4.6, per ciascuna entry all'interno di questa struttura dati comprendono l'identificatore univoco del nodo in questione, la sua posizione, velocità e direzione corrente, l'ultimo istante temporale nel quale i dati a disposizione sono stati aggiornati, l'identificatore dell'eventuale gruppo al quale il nodo può fare attualmente riferimento e tre diversi valori di densità contenuti nella struttura dati `Density`. Per ogni nodo infatti, viene tenuta traccia della sua densità assoluta, ovvero il numero complessivo di nodi che esso vede all'interno del proprio raggio di comunicazione, la densità conformante, cioè quanti tra

questi dispongono di una direzione conformante al nodo stesso ed infine la densità netta che indica quanti nodi fra quelli conformanti non appartengono al momento ad alcun gruppo (cioè sono inattivi).

Per recuperare le informazioni relative ai diversi nodi presenti nelle vicinanze questo modulo sfrutta due fonti differenti: in primo luogo esso analizza costantemente tutto il traffico in entrata al nodo su cui risiede ed in base ad esso ricostruisce e tiene traccia in maniera passiva delle informazioni disponibili per i diversi mittenti incontrati. Per integrare questo processo, specialmente nei periodi di tempo in cui il nodo è inattivo, esso ricorre attivamente al protocollo `PID_HEARTBEAT` per inviare periodicamente messaggi CAM (Cooperative Awareness Message), denominati heartbeat, finalizzati unicamente alla divulgazione delle proprie informazioni di stato ai nodi vicini. Ciascuna di queste due modalità di recupero dati verrà di seguito analizzata nel dettaglio.

Ogni volta che questo componente rileva l'esistenza di un qualsiasi nodo nelle vicinanze e riesce ad ottenere informazioni sul suo stato, aggiorna automaticamente il campo `LastSeen` per l'entry corrispondente, in modo da tenere traccia dell'ultimo istante in cui la presenza di un veicolo è stata rilevata. Se non vengono più ricevute informazioni relative a quel nodo, trascorso un intervallo di tempo configurabile questo viene rimosso dalla struttura dati.

Analisi del traffico in ingresso

Nel paragrafo 3.2.4 è stato illustrato come ogni messaggio scambiato tra due nodi ITS qualunque contiene sempre un header generale (`ItsHeader`) che contiene numerose informazioni sul mittente del pacchetto. In particolare, queste ultime sono elencate nella Tabella 3.2 e comprendono indicazioni sulla posizione, direzione ed identità del nodo mittente. Per sfruttare questa forma di conoscenza implicita il modulo `BehaviourNodeMap` indica al controllore il proprio interesse alla ricezione di qualsiasi messaggio ricevuto, indipendentemente dal particolare `pid`. Al momento della ricezione di un pacchetto, in maniera del tutto trasparente alla logica alla quale esso è realmente destinato, vengono analizzati i diversi campi del suo header principale per ricavare le informazioni utili sul nodo che lo ha trasmesso.

Ulteriori informazioni sullo stato dei mittenti si possono ottenere considerando il `ProtocolId` di ogni messaggio scambiato. Gli header relativi ad alcuni protocolli (in particolare per i messaggi utilizzati per la raccolta di informazioni fra i membri di un gruppo, vedi paragrafo 4.1.2) contengono infatti diverse informazioni utili per dedurre alcune proprietà importanti relative ai nodi vicini. Ad esempio, se si riceve un

Tabella 4.7: Informazioni contenute nei messaggi heartbeat.

Campo	Tipo	Descrizione
Speed	double	Velocità attuale del nodo.
Density	Density	Struttura dati che contiene i valori di densità assoluta, conformante e netta relativi al nodo.

messaggio di leadership da un veicolo si saprà con certezza che esso è il coordinatore di un gruppo appena formato e si potranno quindi estrarre le informazioni sul relativo `GroupId` dall'header del pacchetto.

Questa forma di conoscenza implicita sui nodi circostanti è particolarmente utile in quanto ottenibile in maniera gratuita, semplicemente ascoltando le comunicazioni in corso. In particolare, essa garantisce dei risultati ottimali alla presenza di nodi relativamente vicini ad una RSU e già attivati per un particolare gruppo, visto che i diversi membri dei plotoni scambiano periodicamente numerosi messaggi per il mantenimento del ciclo di vita del gruppo (vedi paragrafo 4.1.2). È tuttavia necessario prevedere un protocollo ausiliario per poter ottenere informazioni anche su nodi che non sono ancora stati attivati.

Scambio di messaggi heartbeat

Un nodo inattivo, in accordo con le linee guida delineate nel paragrafo 4.1.1, resta normalmente silente ed è quindi impossibile da rilevare mediante la tecnica passiva appena esposta. Per ovviare a questo problema si deve necessariamente poter rilassare il vincolo che impedisce ai nodi inattivi di generare traffico di rete, per tollerare l'invio sporadico di semplici messaggi, detti heartbeat, con funzionamento analogo ai Cooperative Awareness Message frequentemente adottati in letteratura e recentemente standardizzati [ETS13].

Nel presente sistema non si farà ricorso ad un particolare standard ufficiale sull'uso dei CAM, ma se ne utilizzerà un semplice formato ad-hoc che contiene le informazioni mostrate nella Tabella 4.7. I campi presenti al suo interno sono complementari a quelli dell'header principale (`ItsHeader`) e permettono complessivamente di ottenere tutte le informazioni utili su un nodo, anche nei periodi di tempo in cui questo è inattivo. È inoltre previsto un messaggio heartbeat particolare per segnalare la volontà di un nodo di essere rimosso dalle mappe dei suoi vicini, ad esempio per segnalare un suo imminente spegnimento.

Il modulo `BehaviourNodeMap` adotta una strategia intelligente per ridurre al minimo il ricorso ai messaggi CAM: quando possibile si considerano le informazioni ottenute passivamente mediante l'ascolto del traffico in ingresso. In linea di principio si prevede l'invio di messaggi heartbeat ad intervalli regolari sufficientemente distanti tra loro per non comportare un eccessivo costo in termini di pacchetti trasmessi. Il periodo di tempo che intercorre tra due invii successivi viene alterato con un piccolo valore casuale per evitare che due o più nodi possano inavvertitamente sincronizzarsi e cancellare vicendevolmente le relative trasmissioni dati. Se il modulo rileva che il proprio controllore ha effettuato l'invio di un qualsiasi messaggio relativo ad un altro protocollo, l'intervallo che precede l'invio dell'heartbeat successivo viene resettato. In questo modo è possibile evitare l'invio di pacchetti CAM per quei nodi che sono comunque impegnati in una conversazione e che quindi forniscono informazioni circa la propria presenza in via indiretta. Non appena il nodo rimane silenzioso per un tempo sufficientemente lungo, il protocollo di heartbeat riprende ad operare normalmente.

Il componente appena descritto garantisce una buona versatilità al sistema in quanto può essere utilizzato da qualsiasi logica aggiuntiva ed il protocollo heartbeat può essere esteso per includere ulteriori informazioni oltre a quelle attualmente previste. Il suo utilizzo relativo ai nodi inattivi deve però essere considerato con attenzione: l'invio costante nel tempo di CAM comporta un traffico di rete non sempre giustificato e le trasmissioni di tali pacchetti potrebbero interferire con le normali attività dei nodi circostanti in quanto aggiungono una fonte di messaggi che aumenta la possibilità di creare collisioni, specialmente in ambienti con elevata concentrazione di veicoli. Il limitato, ma costante traffico generato per mezzo del protocollo heartbeat può inoltre aumentare il consumo energetico di quei veicoli che sfruttano dispositivi con autonomia limitata per partecipare al sistema (ad esempio nel caso di utilizzo di smartphone). Nel corso dell'analisi dei test effettuati verrà esaminato l'effetto che i messaggi CAM hanno sulle performance complessive del sistema realizzato.

4.1.2 Protocolli di gestione del ciclo di vita

Nel corso del paragrafo 2.2.2 sono state ampiamente discusse le funzionalità che un gruppo deve realizzare per informare adeguatamente la RSU a cui fa riferimento dello stato del traffico per la propria direzione, così come viene complessivamente percepito per mezzo dei suoi membri. Ciascun componente del plotone deve pertanto raccogliere quante più informazioni possibili sul suo stato e su quello dei

nodi circostanti ed inviarle periodicamente al proprio leader che provvederà a sintetizzarle ed a spedire un resoconto complessivo dei dati a sua disposizione alla propria RSU di riferimento.

Compito secondario, ma altrettanto importante, del leader di un gruppo è quello di monitorare costantemente il proprio stato e quello dei propri nodi membri: occorre tenere sempre presente che i partecipanti di un plotone, incluso il leader stesso, corrispondono a veicoli dotati di una propria mobilità. Anche se i processi di formazione del gruppo sopra descritti si concludono normalmente in un tempo relativamente breve (circa $500 \div 800$ ms nella configurazione adottata per i test) tale per cui, con le dovute approssimazioni, i nodi che vi partecipano possono essere considerati pressoché fermi, il ciclo di vita di un gruppo può durare da alcuni secondi ad un paio di minuti. Nel corso di questo intervallo di tempo i vari nodi che lo compongono possono andare incontro a disconnessioni, guasti, cambiamenti di direzione ed allontanamenti reciproci. Anche il leader è suscettibile delle stesse condizioni che possono portare un qualsiasi nodo ad uscire dal gruppo. Il sistema prevede inoltre che eventuali nodi inattivi che si trovino a transitare nelle vicinanze di un nodo leader possano decidere autonomamente di entrare a far parte del relativo gruppo, senza dover ricorrere ai protocolli di formazione descritti all'inizio del presente capitolo.

Un gruppo è quindi un'entità che subisce continue trasformazioni ed i cui membri possono variare in qualsiasi momento. Spetta dunque al suo coordinatore controllare periodicamente quanti e quali nodi vi stanno partecipando ed ottenere da questi informazioni utili per lo svolgimento del suo compito. I diversi messaggi scambiati tra leader e membri che consentono l'esecuzione del protocollo di gestione dei nodi di un gruppo rientrano all'interno del `pid` rappresentato dal valore `PID_GROUP_MANAGEMENT` e comprendono vari messaggi che verranno presentati nei paragrafi successivi.

La logica che definisce il comportamento di un nodo leader è contenuta nel modulo `BehaviourGroupLeader`, mentre quella relativa ad un generico membro di un gruppo viene formalizzata dal modulo `BehaviourGroupMember`. Questi comportamenti sono normalmente inattivi all'interno del sistema e vengono attivati automaticamente nel momento in cui un nodo entra a far parte di un gruppo per poi tornare inerti alla sua uscita. Uno dei compiti principali del componente `ItsGroupController` è quello di occuparsi dell'attivazione e della sospensione dei comportamenti responsabili della gestione del ciclo di vita di un gruppo esposti di seguito.

Tabella 4.8: Informazioni contenute nei messaggi di probe scambiati fra i membri di un gruppo.

Campo	Tipo	Descrizione
Group	Group	Struttura dati che contiene l'identificativo e la direzione del gruppo.
Interval	intero	Durata del periodo di attesa nel quale il leader attende le risposte dei nodi membri.
NumNodes	intero	Numero di nodi attualmente noti al mittente.
Speed	double	Velocità attuale del mittente.
Density	Density	Struttura dati che contiene i valori di densità relativi al nodo.

4.1.2.1 Esplorazione dei nodi membri

La procedura con la quale il leader effettua periodicamente una scansione completa dei membri del proprio gruppo viene chiamata probe ed ha il duplice scopo di recuperare i vari dati riguardanti lo stato dei singoli nodi e di verificare quanti tra loro risultano ancora presenti. Al termine di ogni esplorazione, il leader calcola una sintesi di tutti i dati raccolti e la invia alla propria RSU di riferimento.

La procedura di probe ha inizio nel momento in cui il leader del gruppo invia il messaggio corrispondente e si mette in attesa delle varie risposte dei propri membri, entro un tempo limite da lui prefissato (probe time). Ogni nodo che risponde al segnale di probe nel tempo previsto conferma la propria presenza ed allo stesso tempo consegna le informazioni più recenti in proprio possesso. Sia i messaggi di richiesta spediti dal leader, sia quelli di risposta inviati dai diversi membri contengono le stesse informazioni mostrate nella Tabella 4.8.

Il ricevente di tali messaggi è quindi in grado di ottenere una vasta serie di informazioni sul mittente. Ogni partecipante del gruppo, incluso il leader stesso, riutilizza il componente `BehaviourNodeMap` e delega automaticamente ad esso la costruzione e la gestione della mappa dei nodi comprensiva per i diversi membri. Dal momento che i messaggi probe vengono scambiati con più frequenza rispetto a quelli previsti per il protocollo di heartbeat, tale componente ricorrerà esclusivamente all'analisi del traffico in ingresso ed uscita per tenere traccia dei diversi veicoli.

Una volta terminato il tempo stabilito per l'esplorazione dei diversi nodi, il leader verifica se la RSU relativa al gruppo è presente nella propria mappa dei nodi ed

Tabella 4.9: Informazioni contenute nei messaggi di inviati dal leader di un gruppo alla sua RSU di riferimento.

Campo	Tipo	Descrizione
Group	Group	Struttura dati che contiene l'identificativo e la direzione del gruppo.
GroupStats	GroupStatistics	Struttura dati contenente le informazioni di stato destinate alla RSU.
IsDispodes	booleano	Flag che indica se il gruppo è stato chiuso in seguito all'ultima iterazione.

in caso affermativo provvede a calcolare una media delle informazioni appena raccolte ed a trasmetterle in un messaggio di informazione sul gruppo. Questo messaggio viene inviato solo se la relativa RSU è all'interno del range del leader. In seguito ai salti di attivazione può infatti verificarsi la formazione di un gruppo in cui alcuni o tutti i suoi membri risiedono all'esterno del campo di azione della RSU. Raccogliendo informazioni dai diversi leader attivi nelle varie direzioni di interesse, ogni RSU è in grado in questo modo di monitorare l'andamento nel tempo dei vari parametri utili a comprendere le condizioni attuali del flusso del traffico in avvicinamento. Le informazioni contenute in questi messaggi sono mostrate nella Tabella 4.9.

Le esatte tempistiche utilizzate per scandire e regolare la fase di esplorazione devono essere considerate attentamente al fine di scongiurare una scarsa resa dell'algoritmo e di conseguenza dell'intero sistema. Specialmente negli scenari caratterizzati da un'elevata densità, i gruppi tendono ad essere composti da numerosi membri i quali si trovano a loro volta ad operare in un ambiente molto rumoroso, ovvero dove esistono molti altri nodi nelle vicinanze che partecipano a comunicazioni all'interno di altri gruppi. Le immediate vicinanze di una qualsiasi RSU risultano essere delle aree particolarmente critiche non solo per il potenziale sovraffollamento di veicoli ma anche per l'elevato numero di potenziali gruppi contemporaneamente presenti e la conseguente quantità di trasmissioni di dati che vengono continuamente scambiate, prevalentemente a causa del protocollo di raccolta dati utilizzato dai molteplici gruppi.

Per queste motivazioni, è di fondamentale importanza che ogni nodo adotti alcune strategie per tentare di minimizzare la possibilità di incorrere in interferenze tra i pacchetti trasmessi e cercare comunque di mantenere adeguate performance anche nel caso di perdita di alcuni messaggi. Una prima semplice precauzione utilizzata è quella di evitare di rimuovere un nodo dalla lista dei membri conosciuti a seguito di una sola mancata risposta ad un probe del leader: una soglia configurabile del modulo

leader permette di tollerare la mancata risposta di un nodo ad un certo numero di esplorazioni; se il membro in questione continua a non rispondere per un numero di iterazioni superiore al valore specificato, esso viene in ogni caso rimosso.

Una seconda importante strategia prevede il dimensionamento in maniera dinamica del probe time, ovvero del tempo limite concesso all'insieme di nodi che compongono un gruppo per rispondere ad una richiesta ricevuta dal leader. Lo scopo è quello di concedere un intervallo di tempo tanto più ampio quanto maggiore è il numero di nodi presenti nel gruppo. A seguito della ricezione di un messaggio proveniente dal leader, ciascuno di essi è infatti tenuto a scegliere un istante di tempo casuale con distribuzione uniforme tra zero ed il valore specificato nel campo `Interval` dell'header ricevuto. Se il numero di membri è particolarmente alto, dilatare tale intervallo diminuisce la probabilità che i diversi pacchetti si sovrappongano cancellandosi a vicenda. Nel momento in cui un leader intende dare inizio ad un'iterazione di probe, calcola il valore dell'intervallo in modo tale che abbia un valore compreso fra un valore minimo ed uno massimo configurabili e che aumenti progressivamente con il numero di veicoli presenti all'interno del gruppo. Si è scelto di mantenere il risultato asintoticamente lontano dal valore scelto come massimo in quanto si decide di prevedere un periodo di pausa tra due iterazioni di esplorazione consecutive.

Un ultimo accorgimento relativo alle temporizzazioni del protocollo riguarda l'intervallo di tempo che separa la fine di un'iterazione dall'inizio di quella successiva: per evitare che diverse procedure di esplorazione tendano a sovrapporsi si sceglie di sommare a questo intervallo un valore scelto attraverso una distribuzione uniforme e delimitato in modulo ad un valore massimo configurabile. La durata della separazione fra due esplorazioni, a meno della distorsione introdotta, sarà calcolata come la differenza fra il tempo di probe massimo configurato e l'intervallo di tempo riservato all'esplorazione ottenuto precedentemente.

Dal punto di vista dei singoli membri del gruppo la procedura di esplorazione si svolge in maniera elementare: essi provvedono a rispondere al messaggio di probe inviato dal loro leader con un messaggio di risposta dello stesso tipo contenente le informazioni di stato più recenti in loro possesso. Come è già stato accennato, ogni nodo risponde scegliendo un istante di invio casuale uniformemente distribuito all'interno dell'intervallo temporale specificato dal leader. Questo messaggio è trasmesso in geobroadcast per fare in modo che anche i membri del gruppo possano costruire

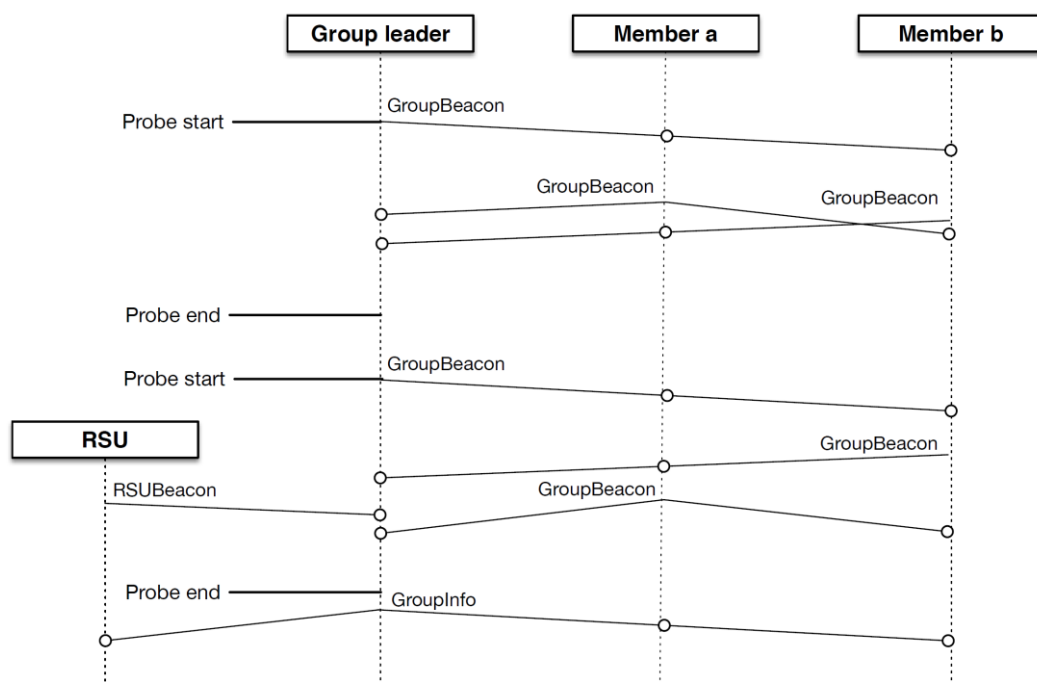


Figura 4.4: Diagramma della sequenza dei messaggi scambiati tra i membri di un gruppo durante due procedure di esplorazione consecutive. Al termine della seconda iterazione, il leader provvede all'invio del messaggio di informazioni in quanto la RSU di riferimento è entrata all'interno del suo range.

senza costi aggiuntivi³ una mappa parziale degli altri loro pari. Questa interazione può avvenire solamente se il nodo ha verificato di possedere una direzione ancora conforme con quella prevista per il gruppo a cui appartiene. Nel caso il veicolo inizi a seguire un percorso inappropriato esso si disattiverà autonomamente, evitando di rispondere al proprio leader.

Ogni volta che viene ricevuto un messaggio di probe proveniente dal proprio leader, ciascun membro del gruppo innesca un timer di durata configurabile; se allo scadere del tempo fissato non sono pervenute ulteriori comunicazioni dal gestore del gruppo, esso considera persa la comunicazione col proprio leader e ritorna a uno stato di completa inattività.

³ A livello fisico una trasmissione wireless unicast o broadcast è identica per quanto riguarda l'utilizzo di banda.

Tabella 4.10: Informazioni contenute all'interno della struttura dati `GroupStatistics` ricevuta dalla RSU.

Attributo	Tipo	Descrizione
Members	intero	Numero di membri del gruppo.
AvgSpeed	double	Velocità media dei membri del gruppo.
MaxSpeed	double	Velocità massima di un membro del gruppo.
Diameter	double	Dimensione del diametro del gruppo.
AvgDensity	double	Densità conformante media dei membri del gruppo.

Il diagramma mostrato in Figura 4.4 illustra un esempio nel quale tre nodi appartenenti allo stesso gruppo comunicano tra loro durante due iterazioni di probe consecutive. Al termine della prima il leader del gruppo non ha ancora la RSU all'interno del proprio range e pertanto conclude la procedura senza ulteriori comunicazioni. Nel corso della seconda iterazione il leader viene avvistato in seguito alla ricezione di un beacon; trovando la RSU di riferimento all'interno del proprio raggio operativo, il coordinatore provvede alla consegna di una elaborazione delle informazioni appena ottenute.

Le informazioni che ogni RSU riceve dai gruppi a cui fa riferimento sono rappresentate dalla struttura dati `GroupStatistics` e sono mostrate nella Tabella 4.10. Il diametro di un gruppo consiste in una stima dell'estensione di un plotone in un dato momento ed è definito come la distanza che separa il membro più lontano dalla RSU di riferimento dal nodo più vicino; nei gruppi composti da un singolo nodo il diametro assume sempre il valore zero.

4.1.2.2 Terminazione di un gruppo

Esistono due modalità con le quali il gruppo può cessare la propria attività: la prima viene detta terminazione (*dispose*) ed costituisce il normale metodo con cui un gruppo conclude il proprio ciclo di vita quando i suoi servizi non sono più necessari; la seconda è sintomo di una qualche anomalia e si verifica quando, a causa di molteplici motivi, il plotone arriva a perdere tutti i suoi componenti. Esistono infatti diverse condizioni, che saranno discusse nel paragrafo 4.1.2.4, che possono portare un qualsiasi membro di un gruppo a distaccarsi da esso ed a tornare inattivo.

In calce al paragrafo 2.2.2 è stata discussa la condizione che porta un qualsiasi gruppo a cessare la propria attività: nel momento in cui un gruppo è sufficientemente vicino alla RSU di riferimento i diversi veicoli inizieranno, se è loro possibile, ad

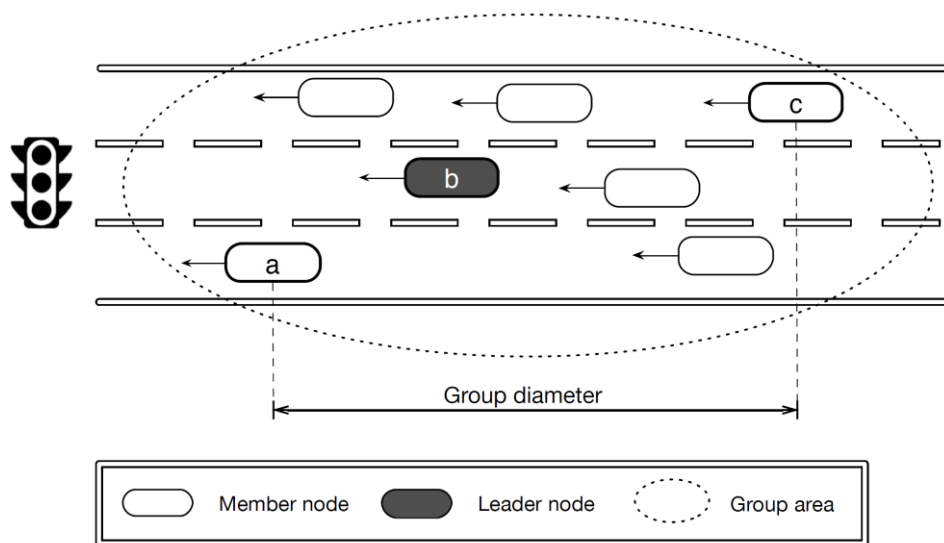


Figura 4.5: Posizioni all'interno di un gruppo: i nodi a e c individuano rispettivamente il punto più vicino e lontano del gruppo dalla RSU; dalla differenza di tali posizioni si ricava il diametro del gruppo. La posizione del nodo leader b viene utilizzata come punto di riferimento per la chiusura del gruppo.

abbandonare la direzione per la quale il gruppo è stato formato e percorreranno strade diverse, dividendosi. In queste situazioni non è più possibile ottenere informazioni utili dai componenti del plotone che deve essere pertanto terminato.

Normalmente non è semplice assegnare una posizione precisa ad un gruppo, considerando che esso si compone di veicoli ciascuno dotato di una propria posizione e di un proprio pattern di mobilità. Esistono comunque diversi punti notevoli che possono essere utilizzati per comprendere con la giusta approssimazione la collocazione e la disposizione di un plotone; essi vengono illustrati nella Figura 4.5 e consistono nel nodo più vicino alla RSU, quello più lontano e quello leader. I primi due tra questi vengono chiamati punti di frontiera e sono utilizzati per ricavare il diametro del gruppo.

Esistono buoni motivi per non considerare i nodi di frontiera come riferimento per la chiusura del gruppo. Se si terminasse un plotone non appena il nodo frontale, ovvero quello più vicino alla RSU, supera una soglia preimpostata si rischierebbe di cessare l'attività del gruppo troppo prematuramente; i membri risulterebbero infatti ancora relativamente lontani dalla RSU e potrebbero contribuire ancora alla raccolta di informazioni utili. Viceversa, chiudendo un gruppo nel momento in cui il suo ultimo nodo, quello più lontano, supera tale soglia, gli altri membri potrebbero avere già superato l'incrocio ed avere abbandonato il gruppo, specialmente il gestore.

Avendo progettato le diverse logiche di formazione in modo che eleggano al ruolo di leader quel nodo che risulta essere approssimativamente al centro del gruppo stesso, si è deciso di semplificare il problema e di considerare la posizione del leader come riferimento per determinare il momento in cui l'attività del plotone deve cessare. Il processo di chiusura di un gruppo viene quindi avviato dal suo leader il quale se, al termine di una procedura di esplorazione, ha rilevato che la propria posizione ha superato una soglia configurabile⁴ invierà un messaggio di informazioni sul gruppo nel quale il flag `IsDisposed` risulterà abilitato (vedi Tabella 4.9). Siccome questi pacchetti sono sempre inviati in geobroadcast sia la RSU, sia i membri del plotone capiscono che il gruppo ha appena completato il suo ultimo probe ed ha cessato la propria attività.

È infine previsto un semplice messaggio di chiusura per forzare manualmente ed in maniera del tutto asincrona⁵ la terminazione di un gruppo. Non è utilizzato da nessuna delle logiche realizzate e la sua implementazione è lasciata ad eventuali estensioni del sistema. Se il leader di un gruppo riceve tale messaggio provvede a ritrasmetterlo ai propri membri per assicurarsi che esso venga ricevuto da tutti i componenti.

4.1.2.3 Annessione ad un gruppo

A causa dell'elevata mobilità dei nodi nel corso della simulazione, è necessario prevedere delle modalità aggiuntive mediante le quali i nodi possano entrare a fare parte di un gruppo quando questo è già formato ed attivo.

Il comportamento che si occupa di questa funzionalità è definito dal modulo `BehaviourGroupJoin` ed il suo funzionamento risulta estremamente semplice. Questo componente svolge la sua funzione nei momenti in cui un nodo è in stato di inattività e viene automaticamente disabilitato all'ingresso in un plotone; la sua attivazione e spegnimento viene gestita in maniera trasparente dal modulo `ItsGroupController`. Quando questo comportamento è in funzione esso si limita a monitorare eventuali messaggi in ingresso appartenenti al protocollo `PID_GROUP_MANAGEMENT` alla ricerca di messaggi di probe provenienti da un gruppo che abbia una direzione di riferimento conformante alla propria.

⁴ È uno degli attributi del componente `ItsGroupController`. Si faccia riferimento all'Appendice A per una lista completa degli attributi dei vari moduli che compongono il sistema.

⁵ La terminazione normale alla fine di un'operazione di probe può essere intesa come sincrona rispetto alle fasi del protocollo.

In questo modo, se un nodo inattivo capta un segnale di probe emesso da un leader⁶ il cui gruppo si sposta seguendo una direzione compatibile, esso potrà inserirsi in tale plotone in qualità di membro. Per comunicare la propria annessione al gruppo il componente di join, quando vengono soddisfatte le condizioni appena esposte, provvede a rispondere al probe inviando un messaggio con i propri dati, seguendo quindi lo stesso comportamento messo in atto dai membri già presenti in un gruppo durante la normale procedura di esplorazione. Il ritardo con cui il componente risponde ad un probe compatibile è scelto casualmente tra un intervallo minimo e massimo configurabile.

In maniera del tutto analoga alla procedura appena descritta, un leader di un gruppo vuoto (ovvero costituito dal solo gestore) può rinunciare al proprio ruolo ed unirsi ad un altro gruppo percepito all'interno del proprio range. Questa funzionalità è particolarmente utile per evitare che in alcune condizioni possano crearsi tanti gruppi vuoti per una stessa direzione di marcia. Per evitare che due nodi leader possano decidere simultaneamente di cambiare gruppo ed annullarsi a vicenda, si stabilisce un criterio in base al quale, data una coppia di leader ciascuno appartenente ad un gruppo vuoto, esiste una sola configurazione risultante possibile come esito di una loro unione: un nodo gestore può entrare in un altro gruppo compatibile solo se il relativo leader possiede un identificativo (`NodeId`) maggiore del suo. La tecnica appena presentata è chiamata *group merging* e viene gestita dal componente `Behaviour-GroupLeader`. Può essere inibita attraverso un flag nella configurazione.

4.1.2.4 Uscita da un gruppo

Un nodo attivo può uscire dal gruppo a cui appartiene quando questo viene chiuso per azione diretta del leader oppure quando non sono più verificate le condizioni di appartenenza al gruppo definite nel paragrafo 2.2.2. Volendole riassumere, esse stabiliscono che un nodo può fare parte di un gruppo soltanto se esso è in grado di comunicare direttamente con il leader e se il suo percorso mantiene una direzione conforme a quella prevista per il plotone. Quando anche solo uno di questi due vincoli viene a mancare, il nodo deve necessariamente uscire dal gruppo. Si noti che il nodo coordinatore del gruppo vede sempre soddisfatta la prima delle due condizioni.

Nell'eventualità che un nodo si trovi costretto a lasciare un gruppo si preferisce procedere a una sua uscita silenziosa, cioè si evita la necessità di inviare un apposito

⁶ I segnali di probe provenienti dai membri vengono ignorati: come definito nel paragrafo 2.2.2 i nodi membri devono poter comunicare con il leader del gruppo.

Tabella 4.11: Informazioni contenute nel messaggio di handoff inviato da un leader alla sua uscita dal gruppo.

Campo	Tipo	Descrizione
Group	Group	Struttura dati che contiene l'identificativo e la direzione del gruppo.
NodeMap	NodeMap	Struttura dati contenente la mappa dei nodi che leader uscente.

messaggio per segnalare al leader la sua dipartita. Un veicolo che intende abbandonare un gruppo torna semplicemente ad uno stato di inattività; dopo pochi secondi il leader provvederà in ogni caso a rimuoverlo dalla propria mappa dei nodi, in seguito ai ripetuti fallimenti riscontrati nella procedura di probe. Questa scelta comporta purtroppo una minore reattività del leader nel rilevare della perdita dei suoi membri ma è motivata dall'esigenza di ridurre il più possibile il traffico di messaggi tra i diversi componenti di un gruppo, utile soprattutto quando questi si trovano ad operare in aree molto dense di nodi. Se si ha l'esigenza di segnalare attivamente la dipartita di un veicolo da un gruppo è previsto l'utilizzo di un semplice messaggio di leave il quale segnala ai riceventi l'intenzione del mittente di abbandonare il gruppo indicato.

Per coordinare meglio il comportamento dei veicoli, si stabilisce un preciso momento nel quale essi devono verificare la loro conformità alla direzione del gruppo. Tale istante viene fatto coincidere con l'inizio di un'iterazione della procedura di esplorazione dei nodi. Come è già stato accennato, ogni membro effettua un test sulla propria direzione alla ricezione del messaggio di probe inviato dal proprio leader ed in caso di fallimento si disattiva e non risponde all'appello. Il gestore del plotone verifica invece la propria ammissibilità prima di inviare tale messaggio.

L'uscita del leader dal gruppo è un evento abbastanza problematico in quanto comporta la perdita dell'intero gruppo e di tutte le informazioni in esso raccolte. Per rispondere a questo problema viene implementato un protocollo di handoff mediante il quale il leader, prima di cessare definitivamente la propria attività, cede il controllo del gruppo ad un nodo membro scelto in base al suo valore di densità. Tale procedura si articola in due semplici passaggi ed ha inizio con l'invio da parte del leader uscente di un messaggio di handoff destinato esclusivamente al nodo da lui scelto come successore. Compiuto questo primo passo, il mittente si disattiva ed abbandona il gruppo senza attendere una conferma dal proprio potenziale successore per non aumentare eccessivamente il costo del protocollo. Le informazioni inviate con questo messaggio sono mostrate nella Tabella 4.11.

Tabella 4.12: Informazioni contenute nel messaggio inviato dal nuovo leader del gruppo per completare la procedura di handoff.

Campo	Tipo	Descrizione
Group	Group	Struttura dati che contiene l'identificativo e la direzione del gruppo originale.
NewGid	intero	Nuove identificatore per il gruppo.

Un messaggio di handoff contiene l'intera mappa dei nodi costruita fino a quel momento dal gestore originale e permette in questo modo al successore di riprendere la guida del gruppo senza perdite di dati. Per come è stato strutturato il sistema⁷, il cambiamento di un leader comporta necessariamente la modifica del `gid` del gruppo per fare sì che sia i membri, sia la RSU di riferimento riconoscano il nuovo leader come tale. Per informare i nodi di un gruppo della necessità di effettuare il passaggio a un nuovo leader, il nuovo gestore del gruppo completa l'handoff con un messaggio di cambiamento del gruppo le cui informazioni sono mostrate nella Tabella 4.12.

4.2 Protocollo centralizzato

Come per ogni nuova tecnologia, anche l'introduzione delle comunicazioni V2X all'interno dei veicoli avverrà in modo graduale, partendo da tassi di penetrazione molto bassi, nell'ordine della singola cifra percentuale, e passando progressivamente ad una diffusione sempre maggiore. Nell'ambito della raccolta di informazioni sul flusso del traffico è quindi di fondamentale importanza lo sviluppo di protocolli che possano fornire una buona approssimazione del reale stato del sistema anche se la diffusione delle tecnologie su cui si basano è molto limitata. Le prestazioni di questi protocolli possono favorire la diffusione delle tecnologie stesse, infatti se i vantaggi derivanti dal loro utilizzo sono da subito evidenti si incentivano gli investimenti volti a favorirne la loro diffusione.

Come introdotto nel paragrafo 2.3 questo protocollo è stato sviluppato con l'obiettivo di migliorare le prestazioni nelle sistema negli scenari caratterizzati da basse penetrazioni. Nei due paragrafi seguenti verrà illustrato come avviene la comunicazione fra i nodi del sistema e quali dati sono raccolti.

⁷ Vedi definizione di `GroupId` al paragrafo 4.1.1.

4.2.1 Comunicazione fra i nodi

Focalizzandosi sugli scenari nei quali il numero di veicoli equipaggiati con le tecnologie di comunicazione utilizzate in questo progetto è ridotto rispetto al numero totale di nodi si è cercato di sviluppare un protocollo ottimizzato alla raccolta di informazioni sullo stato del traffico in modo che fosse il più semplice ed estendibile possibile.

Analizzando i risultati ottenuti attraverso il protocollo decentralizzato dettagliato in precedenza si è rilevato che il numero medio di membri all'interno di un gruppo varia in modo approssimativamente proporzionale al variare della penetrazione della tecnologia utilizzata⁸. Questo risultato è in linea con le aspettative e non dipende dalla particolare implementazione o da possibili difetti nella procedura di formazione dei gruppi del protocollo utilizzato. Il concetto di gruppo è stato introdotto per ottimizzare la comunicazione fra veicoli e per migliorare la raccolta dei dati, ma il suo scopo viene a mancare quando il numero di membri che lo compongono è molto ridotto. L'utilità di raggruppare i singoli nodi in plotoni si riduce negli scenari nei quali è presente una scarsa penetrazione a causa dell'overhead comunque presente per la formazione e per la gestione del suo ciclo di vita. Si è quindi deciso di eliminare completamente questo concetto durante la definizione di questo nuovo protocollo.

La decisione di eliminare il raggruppamento dei veicoli può comportare la creazione di una comunicazione disorganizzata fra i nodi, che, essendo indipendenti e non più controllati da un gestore, potrebbero trasmettere una gran quantità di messaggi causando problemi di intasamento con il conseguente aumento delle collisioni, da cui deriva una riduzione delle prestazioni del protocollo, nel caso la concentrazione dei veicoli superi una certa soglia, che potrebbe essere anche abbastanza contenuta nei casi peggiori. Per porre rimedio a questo problema si è deciso di introdurre forti limitazioni sulle possibilità di comunicazione di un nodo con l'ambiente circostante.

Il primo vincolo che ci si è dati per ridurre il problema descritto consiste nella riduzione dei possibili destinatari, eliminando le comunicazioni dirette fra i veicoli che vengono limitati allo scambio di dati con i dispositivi infrastrutturali (V2I). Si è infatti ritenuto superfluo lo scambio di messaggi fra i veicoli durante lo sviluppo del protocollo, avendo deciso di non adottare politiche di routing per i messaggi e di limitare quindi la raccolta dei dati all'interno del solo raggio di copertura di una RSU.

⁸ Maggiori dettagli saranno proposti nel paragrafo 5.4.6.

Tabella 4.13: Informazioni contenute nei messaggi beacon inviati dalle RSU.

Campo	Tipo	Descrizione
RSUId	intero	Identificatore della RSU che richiede la formazione del gruppo.
Direction	double	Singolo valore di direzione al quale si riferisce il messaggio corrente.
MaxResponse	intero	Intervallo di tempo entro il quale la RSU aspetta le risposte dai nodi.

Il secondo vincolo riguarda la logica di scambio di messaggi dei nodi: per reintrodurre una figura di gestore che si occupa di coordinare e di controllare le comunicazioni si è deciso di vietare la trasmissione di messaggi da parte dei veicoli se questa non viene esplicitamente richiesta. I veicoli rimangono quindi in uno stato di inattività in assenza di stimoli ricevuti dall'esterno; solo quando ricevono un messaggio da parte di un dispositivo infrastrutturale possono trasmettere una risposta, dopo aver verificato che alcune condizioni siano soddisfatte. In seguito alla ricezione del messaggio e l'eventuale invio di una risposta, il nodo ritorna nello stato inattivo. In questo modo alle RSU viene dato il completo controllo sulle comunicazioni dei nodi che partecipano alla raccolta dei dati.

I messaggi inviati da un dispositivo infrastrutturale per sollecitare lo scambio informazioni sono dei beacon che, similmente a quelli introdotti in precedenza protocollo decentralizzato, contengono le direzioni alle quali è interessata la RSU. Quando un nodo riceve tale messaggio controlla se la propria direzione attuale è conforme, cioè se è compresa all'interno di un certo angolo di tolleranza, ed in caso affermativo procede ad inviare una risposta. Vi sono due differenze però rispetto ai messaggi beacon introdotti in precedenza: questi contengono una singola direzione ed indicano al nodo un intervallo di tempo entro il quale rispondere. Le informazioni contenute in questi messaggi sono mostrate nella Tabella 4.13.

Le immediate vicinanze di una qualsiasi RSU sono le aree più critiche per il potenziale sovraffollamento di veicoli, soprattutto nel caso di penetrazioni molto elevate, e la conseguente quantità di trasmissioni dati che possono essere avviate con un singolo beacon inviato dalla RSU. Per limitare in modo semplice il numero potenziale di nodi che possono rispondere ad un particolare messaggio si è deciso di permettere la comunicazione ai soli veicoli che procedono in una delle direzioni controllate dalla RSU inserendo un solo valore all'interno della richiesta. Per ottenere informazioni su

tutte le direzioni controllate, queste vengono esplorate in messaggi beacon successivi attraverso una politica di round robin.

Il secondo parametro inviato in un messaggio di beacon dalla RSU indicata l'intervallo di tempo entro il quale i nodi possono rispondere. Alla fine del tempo allocato la RSU provvederà ad una nuova trasmissione per una differente direzione. Un nodo che riceve un messaggio da una RSU controlla se la propria direzione attuale è conforme con quella indicata ed in caso negativo scarta il messaggio. In caso positivo sceglie un ritardo per l'invio del proprio messaggio di risposta casualmente attraverso una distribuzione uniforme in modo che il valore massimo possibile sia quello specificato dalla RSU. È stato deciso di lasciare un piccolo delay, configurabile fra le impostazioni del nodo, fra la trasmissione di un beacon e l'inizio dell'invio delle risposte e fra la fine delle risposte e l'invio di un nuovo messaggio dalla RSU per evitare anche la remota, ma possibile, sovrapposizione di questi messaggi. A tal fine il valore di ritardo calcolato dal nodo è limitato sia inferiormente, a questo delay, che superiormente, al valore indicato dalla RSU sottratto il delay. Trascorso questo periodo il nodo invia la sua la sua risposta previa verifica del controllo di conformità per evitare la trasmissione nel caso la propria direzione si sia discostata da quella prevista. In seguito alla comunicazione il nodo ritorna in uno stato di inattività.

La trasmissione del tempo massimo di risposta all'interno dei messaggi inviati dalla RSU può sembrare superflua ma è stata introdotta per garantire il controllo completo sul protocollo di comunicazione ai dispositivi dell'infrastruttura. Due RSU potrebbero utilizzare una temporizzazione differente senza la necessità di alcun intervento sui singoli nodi. Questo parametro può influire in modo rilevante sulle performance del protocollo, infatti la scelta di un valore troppo breve può portare ad un aumento delle collisioni nel caso in cui molti veicoli vogliano rispondere, viceversa un valore molto lungo riduce la frequenza con la quali vengono raccolte le informazioni.

È inoltre possibile adattare questo parametro sia alle condizioni morfologiche del particolare incrocio, ad esempio per allocare un valore minore in una direzione con una sola corsia ed un tempo maggiore per una strada con più corsie, sia dinamicamente in base alle condizioni istantanee del traffico, ad esempio aumentando il tempo concesso nelle direzioni che al momento sono molto affollate. Nella valutazione di questo protocollo si è scelto di impostare questo parametro staticamente sia perché ci si è concentrati soprattutto sulle prestazioni negli scenari con basse concentrazioni di veicoli nei quali ha poco impatto, sia perché i risultati ottenuti negli altri scenari è rimasta comunque accettabile.

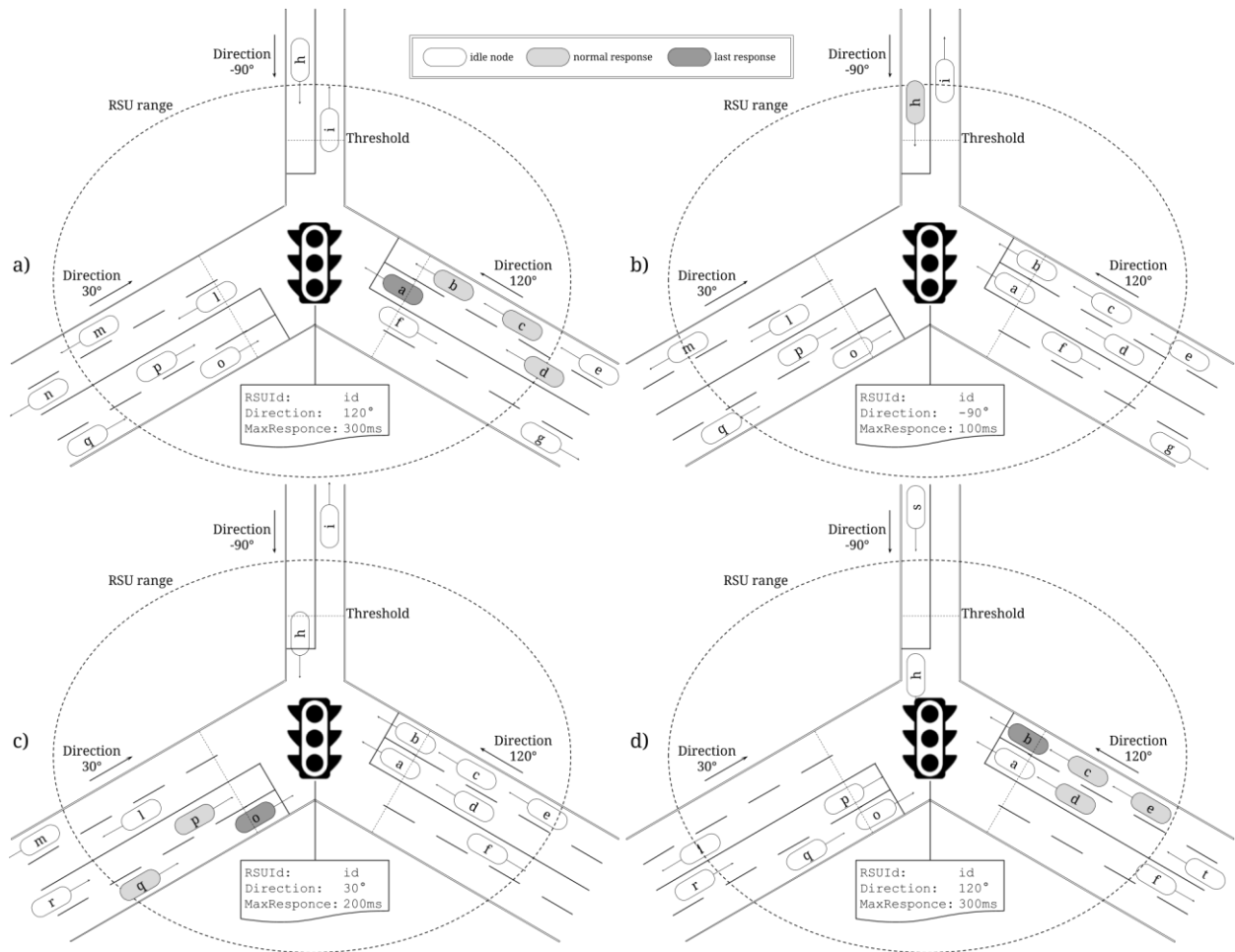


Figura 4.6: Schematizzazione di una serie di messaggi beacon nel protocollo centralizzato. I nodi con direzione 120° rispondono al beacon entro 300ms. Il nodo a invia l'ultimo messaggio avendo superato la soglia configurata (a). Successivamente risponde il nodo con direzione -90° entro 100ms (b). I nodi con direzione 30° devono rispondere entro 200ms. In nodo o invia il suo ultimo messaggio (c). Tornano a rispondere i nodi con direzione 120°. In nodo a ignora il beacon e rimane inattivo (d). In questo esempio i tempi di risposta ai beacon tengono conto della singola corsia nella direzione -90° e del maggiore traffico in direzione 120° rispetto a 30°.

Come introdotto nel paragrafo 2.3, occorre stabilire quando interrompere la trasmissione delle risposte ai beacon da parte di un nodo. Un veicolo dopo essere entrato all'interno dell'incrocio potrebbe comunicare delle informazioni di scaso interesse alla RSU in quanto non dipendenti direttamente dalla condizione del traffico; per questo motivo è utilizzata una soglia configurabile superata la quale il veicolo risponde ad un ultimo beacon in cui indica alla RSU la sua uscita dalla raccolta dati. È inoltre importante evitare che i nodi continuino a rispondere dopo aver superato l'incrocio, è

Tabella 4.14: Informazioni contenute nei messaggi di risposta inviati dai nodi.

Attributo	Tipo	Descrizione
Direction	intero	Direzione attuale del nodo.
Speed	double	Velocità istantanea dell'ultimo campionamento.
AvgSpeed	double	Velocità media su un numero configurabile di campionamenti.
Last	double	Flag che indica l'ultima risposta da parte del nodo alla RSU.

pertanto previsto che, una volta terminata la comunicazione con una particolare RSU, non possano riprenderla fino a quando non ne incontrano un'altra.

In Figura 4.6 è schematizzato il funzionamento completo di questo protocollo con un esempio nel quale i tempi di risposta ai beacon sono modificati dinamicamente per illustrare le possibilità offerte.

4.2.2 Raccolta dati

Dopo aver chiarito la modalità di comunicazione utilizzata dai dispositivi in questo protocollo in questo paragrafo illustreremo quali dati sono trasmessi alle RSU.

Ogni messaggio di risposta ad un beacon inviato da un nodo contiene esclusivamente le informazioni su proprio stato, in quanto questo non ha nessun modo di scambiare informazioni attivamente con i suoi pari ed i dati eventualmente raccolti in modo passivo sono necessariamente conosciuti anche dalla RSU a causa delle limitazioni imposte nella progettazione del protocollo. Le informazioni scambiate all'interno del header utilizzato sono mostrate nella Tabella 4.14. Questi messaggi contengono l'identità del nodo, la sua posizione attuale, il suo tipo, la sua direzione, la sua velocità istantanea e quella media. Occorre ricordare che il controllore provvederà ad inserire automaticamente nel proprio header l'identificatore del nodo, il suo tipo e la sua posizione. È previsto un flag per indicare alla RSU l'ultimo messaggio inviato da un nodo prima di disattivarsi a causa del superamento della soglia configurata. Questa informazione può essere utilizzata dalla logica di raccolta dei dati per calcolare le statistiche sulle comunicazioni con il particolare nodo e per estrarre ulteriori informazioni, come ad esempio una approssimazione sul tempo impiegato dal nodo per attraversare l'incrocio.

Per tenere conto di possibili perdite di pacchetti e della possibilità che un veicolo abbia cambiato direzione o che sia altrimenti impossibilitato a continuare la comunicazione è previsto un timeout che permette alla RSU di eliminare dalla propria struttura dati i nodi che non hanno risposto ad un certo numero di beacon di direzione opportuna e che non avevano ancora raggiunto la soglia di disattivazione informando la RSU.

I dati ricevuti dal una RSU attraverso questo protocollo sullo stato dei veicoli sono innumerevoli e molto maggiori rispetto a quelli ottenuti attraverso il protocollo decentralizzato. Occorre quindi prevedere una logica di elaborazione dei dati che possa fornire una sintesi adeguata alle euristiche di più alto livello dedicate, ad esempio, al controllo del traffico. Questa logica esegue essenzialmente il lavoro del svolto dal nodo leader al termine di un ciclo di esplorazione dei membri del suo gruppo prima di inviare il resoconto alla RSU.

Il protocollo centralizzato ha il vantaggio di permettere una più semplice modifica ed evoluzione di questa logica di elaborazione in quanto è sufficiente cambiare la programmazione dei dispositivi infrastrutturali senza la necessità di appostare alcuna modifica ai dispositivi mobili, tranne nei casi di cambiamento radicale del protocollo di comunicazione. Questo consente anche la validazione di una particolare evoluzione nell'ambiente reale aggiornando una singola RSU di test senza la necessità di apportare modifiche al resto del sistema.

Nel protocollo decentralizzato le possibilità di evoluzione delle euristiche di raccolta dati sono limitate alle informazioni aggregate ricevute dai vari gruppi, le quali possono mancare di dettaglio, soprattutto nel caso dell'introduzione di logiche evolute per sopperire alla scarsa penetrazione della tecnologia utilizzata. Una modifica alle modalità di elaborazione dei dati del nodo leader comporterebbe la modifica di ogni dispositivo mobile rendendo difficile sia il deploy, sia il test del cambiamento. Occorrerebbe inoltre gestire i possibili problemi derivanti da differenti versioni del protocollo presenti contemporaneamente.

Come sarà illustrato in modo approfondito nel prossimo capitolo, per ottenere una buona approssimazione delle condizioni reali del traffico negli scenari con bassa o variabile penetrazione occorre focalizzarsi su informazioni che hanno una bassa dipendenza dalle variazioni del numero di dispositivi che hanno contribuito alla loro raccolta, come ad esempio la velocità di un nodo oppure in tempo che impiega per

superare l'incrocio. È inoltre evidente come in questi scenari è controproducente aggregare le poche informazioni disponibili prima di comunicarle al dispositivo incaricato a prendere decisioni basandosi su queste.

Capitolo 5

Risultati sperimentali

In questo capitolo saranno presentati gli scenari urbani utilizzati per simulare la risposta dell'architettura descritta fino a questo momento e di seguito saranno riportati e commentati i risultati dei test maggiormente significativi che sono stati svolti su di essi in modo da poter valutare il comportamento dei protocolli realizzati e la qualità dei dati prodotti da questi ultimi.

Si cercherà anzitutto di verificare che il sistema risponda adeguatamente agli stimoli offerti dai diversi ambienti predisposti. Verranno a tale scopo commentati alcuni studi condotti per individuare sotto quali limiti e condizioni è possibile avvalersi delle informazioni generate dai due protocolli e si cercherà di capire se queste sono sufficienti per sintetizzare un'euristica soddisfacente per la determinazione dello stato reale del flusso di traffico nelle varie aree di interesse.

Nell'esporre i diversi risultati conseguiti dalle numerose simulazioni eseguite si porrà particolare attenzione nel confrontare le differenze in termini di costi e prestazioni offerte dalle diverse versioni del protocollo, saranno inoltre studiate le differenze fra le strategie di formazione dei gruppi utilizzate nel protocollo centralizzato ed introdotte nel paragrafo 4.1.1.

Prima di passare alla discussione dei risultati ottenuti dalle simulazioni si introducono gli scenari utilizzati per la raccolta dati ed alcune ottimizzazioni apportate alla piattaforma di test per migliorare le prestazioni ottenute dai protocolli presentati.

5.1 Scenari di simulazione utilizzati

Durante le fasi di implementazione e di messa a punto dell'architettura sono stati utilizzati diversi scenari per osservare il funzionamento dei sorgenti prodotti. Per sperimentare correttamente il comportamento del sistema è necessario avvalersi di tracce di mobilità contenenti reti stradali complete ed adeguatamente complesse per ottenere risultati realistici su ambienti urbani con elevato indice di densità.

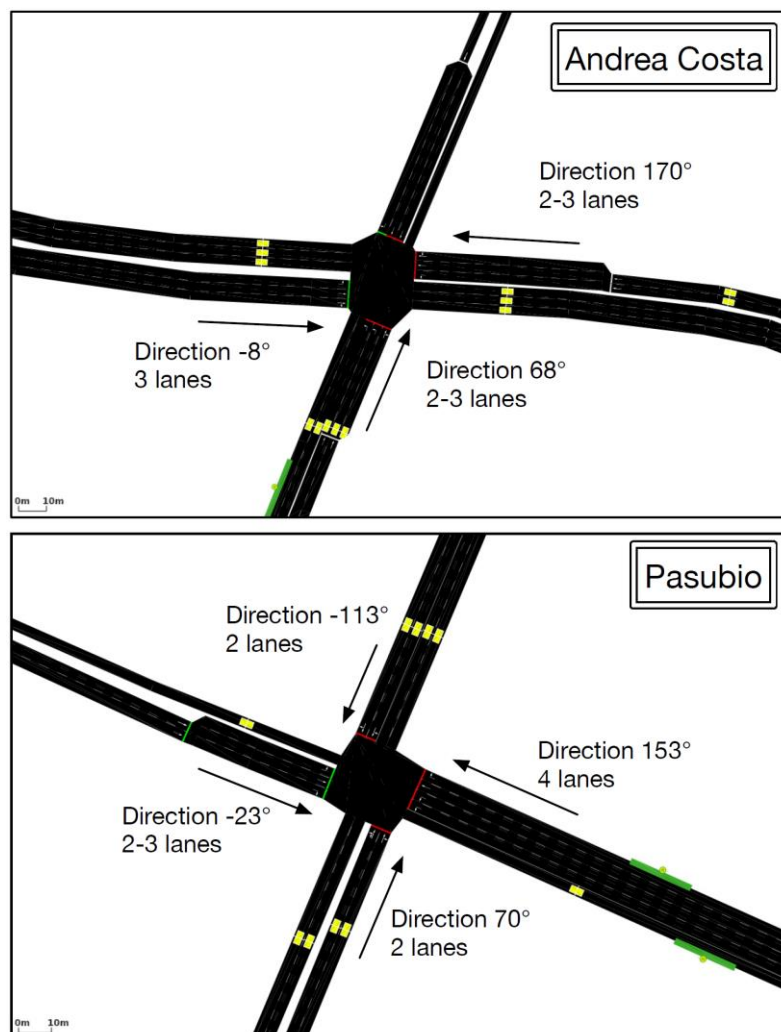


Figura 5.1: Rappresentazione della topologia delle reti stradali di Andrea Costa e di Pasubio degli scenari utilizzati per i test del sistema.

La Figura 5.1 mostra i due scenari utilizzati per produrre i numerosi test presentati in questo capitolo: “Andrea Costa” e “Pasubio”, i quali prendono i loro nomi dalle due strade della città di Bologna da cui sono stati estratti. Entrambi sono stati messi a disposizione rispettivamente dai progetti iTetris e COLOMBO (vedi capitolo 1) e sono stati prodotti a seguito di rilevazioni su flussi reali di traffico effettuate nelle corrispondenti zone della città. Nella figura è possibile osservare le direzioni prese in esame¹ e le caratteristiche dei tratti stradali simulati. Nel complesso, entrambe le

¹ Nello scenario A. Costa la direzione sud-ovest non viene monitorata in quanto la relativa traccia di mobilità non registra il passaggio di alcun veicolo per tale tratto stradale.

tracce di mobilità rappresentano un totale di 48 ore di traffico veicolare. Per ovvi motivi non è possibile presentare in maniera compatta e leggibile i risultati derivati da una simile quantità di dati; nei successivi test vengono pertanto estratti degli intervalli temporali corrispondenti a 30 minuti di simulazione da ciascuna delle due tracce utilizzate. L'intervallo scelto è all'interno del traffico mattutino per ottenere una valutazione del sistema nelle condizioni peggiori.

Le differenze principali tra le due tracce di mobilità utilizzate risiedono nella quantità di nodi simulati e nel numero di corsie che costituiscono i tratti stradali che partono dall'incrocio in esame. Lo scenario A. Costa raggiunge un quantitativo di veicoli all'interno del raggio operativo della RSU che oscilla tra le 60 e le 100 unità; tra le direzioni monitorate, quella con valore -8° risulta la più trafficata con picchi massimi che superano i 50 veicoli e presenta frequenti congestioni. Lo scenario Pasubio presenta una densità decisamente elevata con un traffico a regime che varia fra 80 e 130 veicoli e quattro direzioni caratterizzate da un abbondante flusso di traffico; tra queste il tratto con direzione 153° è sicuramente degno di nota in quanto costituito da quattro corsie che presentano una congestione pressoché permanente all'interno dell'intervallo di simulazione considerato, arrivando a toccare i 70 veicoli nel momento di maggior concentrazione.

5.2 Ottimizzazioni ad iTETRIS

Durante l'implementazione del protocollo decentralizzato ed il successivo lo sviluppo del protocollo centralizzato sono state apportate numerose modifiche al simulatore utilizzato per adattarlo alle particolari esigenze del progetto e per migliorare le prestazioni ottenute. Alcune di queste modifiche sono apparse subito necessarie mentre per altre è stato necessario un lungo lavoro di test per individuare il problema ed implementare una modifica. Si è cercato di creare un'implementazione di queste modifiche il più generale possibile in modo che possano essere riutilizzate in seguito per la creazione di applicazioni differenti rispetto a quelle specificatamente sviluppate per questa tesi. In seguito sono illustrati i cambiamenti più rilevanti apportati alla piattaforma iTETRIS.

L'insieme di queste modifiche ha prodotto miglioramenti molto significativi dei risultati ottenuti, ma ha reso non compatibile la versione utilizzata in questo progetto

con quella standard. In particolare per poter riutilizzare l'interfaccia con iCS² sviluppata in un'altra applicazione occorre utilizzare questa versione modificata della piattaforma. Benché questo possa creare problemi, la decisione di apportare queste modifiche si è resa indispensabile per ottenere risultati accettabili dai protocolli implementati.

Una prima modifica molto importante è stata il cambiamento del tempo dello step di simulazione utilizzato dal simulatore³ durante l'esecuzione. Studiando il protocollo decentralizzato per poi procedere all'adattamento ad iTETRIS è infatti apparso subito completamente inadeguato il valore originario di un secondo e si è quindi proceduto ad apportare le opportune modifiche sia ad iCS sia alla sua interfaccia dedicata all'interazione con i vari componenti del sistema per rendere la risoluzione configurabile e per ridurre il valore minimo ad un millisecondo. Per permettere la massima flessibilità nell'impostazione di questo parametro ne è stata prevista la possibilità di modifica attraverso il file di configurazione utilizzato per avviare una simulazione.

La riduzione dello step di simulazione comporta un aumento del tempo complessivo di simulazione sia a causa del corrispondente incremento dei cicli di iCS necessari a terminarla, sia a causa della maggior quantità di eventi che è possibile generare al protocollo. Durante la fase di test è stata posta particolare attenzione alla scelta del valore ottimale per i vari protocolli utilizzati al fine da ottenere un adeguato compromesso fra le prestazioni ottenute ed il tempo impiegato dal simulatore per completare l'esecuzione. In generale la scelta di una risoluzione minore aumenta le prestazioni del protocollo, ma genera un grande incremento del tempo impiegato per terminare la simulazione. A titolo di esempio passando da 25ms a 5ms si ha un miglioramento molto lieve delle prestazioni a fronte di un tempo di simulazione di quasi tre volte maggiore passando da circa un'ora e mezza a più di quattro ore. Si è scelto di utilizzare il valore di 25ms per il protocollo decentralizzato, mentre per il protocollo centralizzato si è scelta la risoluzione di 50ms, in quanto quest'ultimo si è rilevato meno sensibile alle variazioni di questo parametro. Tutti i dati presentati nel seguito di questo capitolo utilizzano questi valori, salvo diversamente indicato.

Una seconda modifica che ha influito in modo molto significativo sulle prestazioni ottenute dal protocollo è stata una modifica nella modalità di scheduling dei

² Alla descrizione di questo componente è dedicato il paragrafo 3.2.1.

³ Per ulteriori dettagli si faccia riferimento al paragrafo 3.1.4.

messaggi all'interno di NS-3. Confrontano le prestazioni del protocollo decentralizzato originale in NS-3 standard con quelle ricavate dall'implementazione in iTETRIS prima di implementare questa modifica, si ottenevano valori non soddisfacenti ed allo stesso tempo un numero di collisioni di svariati ordini di grandezza superiore. Investigando la causa di questo problema si è infine scoperto che l'implementazione originale dell'interfaccia iNCI⁴ di NS-3 per la comunicazione con iCS prevedeva la schedulazione di tutti i messaggi scambiati dai nodi all'interno di uno step di simulazione all'inizio dello stesso, creando un'enorme congestione con un conseguente grande numero di collisioni.

Questo vanificava qualsiasi tentativo per alleviare questo problema implementata nel protocollo attraverso la scelta casuale dell'istante di invio dei messaggi. Per ovviare al problema si è aggiunta la possibilità per l'applicazione di specificare l'istante d'invio del messaggio all'interno dello step di simulazione successivo in modo tale da avere una simulazione più realistica. Per ottenere questo risultato è stato necessario modificare numerosi componenti: il server di comunicazione implementato in NS-3 e la sua interfaccia con iCS per prevedere la schedulazione dell'invio dei messaggi nell'istante indicato dall'applicazione; la sottoscrizione apposita utilizzata dall'applicazione prevedendo questo valore; adattare iCS perché comunicasse opportunamente questa informazione al simulatore di rete. Dopo aver implementato questa ottimizzazione le prestazioni fra le due implementazioni si sono allineate ed anche il numero di collisioni è sceso a valori comparabili fra le due versioni.

La grande influenza dei cambiamenti della risoluzione e della modalità di schedulazione sui risultati ottenuti è mostrata nella Figura 5.2 per due direzioni nello scenario di Pasubio. Nel grafico è mostrato il numero di vetture rilevato dal protocollo decentralizzato in modalità reattiva con diverse configurazioni: nella versione originale in NS-3 (in nero), in quella con risoluzione di un secondo (in verde), prima della modifica alla modalità di schedulazione dei messaggi (in blu) e nella versione finale che utilizza le due ottimizzazioni precedentemente illustrate (in rosso).

In seguito sono riportate modifiche che hanno avuto un minor peso sui risultati ottenuti, ma che sono nondimeno importanti per la corretta implementazione dei protocolli all'interno di iTETRIS.

⁴ Maggiori informazioni sono fornite nel paragrafo 3.1.2.

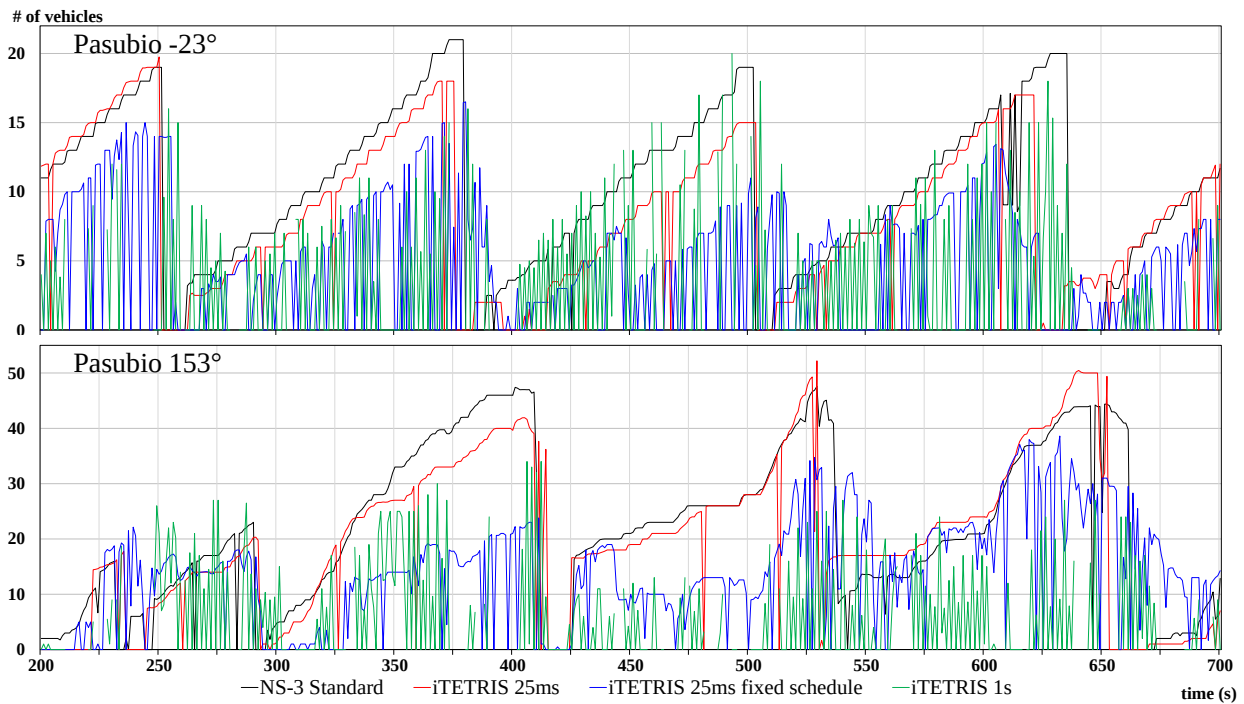


Figura 5.2: Confronto fra varie ottimizzazioni di iTETRIS per il protocollo decentralizzato nelle direzioni -23° e 153° dello scenario di Pasubio. La traccia nera è il risultato ottenuto con la versione originale del protocollo in NS-3, quella verde il risultato ottenuto con la risoluzione di 1s, quella blu rappresenta il risultato ottenuto con 25ms, ma con la schedulazione originale dei messaggi ed infine la traccia rossa contiene i risultati con risoluzione di 25ms e schedulazione dei messaggi modificata come descritto.

La versione standard della piattaforma di simulazione utilizzata supporta la possibilità d'invio di messaggi secondo modalità di comunicazione differenti (fra cui unicast, geobroadcast, topobroadcast, etc.), ma prevedeva l'implementazione della sola modalità unicast per lo scambio di messaggi. I protocolli utilizzati in questo progetto utilizzano esclusivamente la modalità di comunicazione di geobroadcast, eseguendo un eventuale filtraggio dei messaggi a livello applicativo. Per attivare il supporto di questa modalità si sono rese necessarie una serie di modifiche all'interno del solo iCS dato che in NS-3 era stata già prevista questa modalità di comunicazione. L'area di copertura di propagazione è stabilita con due modalità: la prima consiste nelle impostazioni fisiche del dispositivo di trasmissione all'interno di NS-3 (potenza del segnale, guadagno dell'antenna, etc.), mentre la seconda è un raggio di propagazione impostato dall'applicazione attraverso un parametro nella sottoscrizione per l'invio di

messaggi⁵. Un messaggio geobroadcast viene simulato in NS-3 e, nel caso sia ricevuto da un nodo, iCS controlla se il ricevente è all'interno del raggio di propagazione previsto dal mittente ed in caso affermativo consegna il messaggio all'applicazione, altrimenti lo scarta. Il raggio di propagazione massimo è quindi sempre dettato dalle impostazioni del simulatore di rete, mentre quello minimo può essere scelto dall'applicazione.

Per poter funzionare correttamente i protocolli utilizzati devono poter scambiare dati fra nodi differenti, è quindi necessario poter aggiungere un payload ai messaggi. Come è stato illustrato nel paragrafo 3.2.1, per ottimizzare la comunicazione fra l'applicazione ed iCS si è deciso di inviare esclusivamente l'identità di un pacchetto memorizzato localmente, senza la necessità della sua trasmissione completa. Per ottenere questo risultato si è modificata nuovamente la sottoscrizione per l'invio di messaggi in modo che possa contenere questo identificativo, ed il formato utilizzato da iCS per comunicare ad una applicazione la ricezione di un nuovo messaggio. Sono stati inoltre apportati i cambiamenti necessari ad iCS per supportare questo scambio di dati: questo dato non viene comunicato ad NS-3, ma rimane confinato all'interno di questo componente che provvede ad inserirlo nei messaggi ricevuti correttamente prima di notificarli all'applicazione. Cercando di creare un'implementazione il più generale possibile si è deciso di utilizzare come formato per questo payload il tipo stringa di dimensione variabile, in modo tale che possa essere utilizzato non solo per contenere un identificatore intero, ma anche una quantità di dati variabile ed arbitrariamente formattata da una particolare applicazione.

Per migliorare la gestione delle sottoscrizioni è stata prevista la condivisione del loro identificatore fra iTETRIS e le applicazioni. Questa modifica si è resa necessaria in quanto la versione originale di questo framework identifica le sottoscrizioni solamente con il tipo quando procede a richiedere ad una applicazione quali desidera rimuovere. Dato che le applicazioni sviluppate contengono innumerevoli sottoscrizioni di una stessa tipologia (ad esempio nel caso di vari messaggi inviati contemporaneamente) è necessario identificarle correttamente durante la procedura di eliminazione. Per ottenere questo comportamento iCS comunica l'identificativo all'applicazione durante la richiesta, la cancellazione e la conferma⁶ delle sottoscrizioni. In questo modo

⁵ Per semplicità si considera sempre l'area di propagazione di un messaggio circolare.

⁶ Utilizzata ad esempio per confermare la schedulazione di un messaggio nella fase di ricezione dei dati sottoscritti. Maggiori informazioni nel paragrafo 3.1.5.

le applicazioni ed iCS condividono un identificatore univoco permettendo una gestione più accurata della loro vita.

Sono stati infine apportati numerose correzioni ed ottimizzazioni minori all'interno di iCS e dell'estensione di NS-3 con lo scopo di ridurre le risorse necessarie alla simulazione e di velocizzarne l'esecuzione. A questo fine è stata anche introdotta la sottoscrizione illustrata nel paragrafo 3.2.1, volta ad ottimizzare la comunicazione all'applicazione delle posizioni dei nodi presenti nel sistema.

5.3 Configurazione delle simulazioni

Prima di procedere con l'analisi dei risultati verranno brevemente descritte le procedure necessarie per predisporre e configurare l'esecuzione di una simulazione in iTETRIS che utilizzi l'insieme dei moduli ITS prodotti. I due protocolli discussi nel capitolo 4 sono simulati separatamente utilizzando due applicazioni per iCS distinte, entrambe basate sulla stessa interfaccia descritta nel paragrafo 3.2.1. Le configurazioni delle due applicazioni sono uguali per le parti in comune e si differenziano esclusivamente nelle impostazioni dei comportamenti specifici al particolare protocollo. In particolare il modulo dedicato alla simulazione della mobilità, descritto nel paragrafo 3.2.2, è configurato nello stesso modo per ottenere un comportamento dei veicoli comparabile nei due sistemi.

Per eseguire un'istanza di simulazione del sistema realizzato attraverso iTETRIS occorre predisporre e configurare una serie di file che saranno utilizzati per definire il comportamento dei quattro moduli principali che costituiscono il sistema: il componente iCS, i simulatori SUMO ed NS-3 e l'applicazione che contiene i protocolli simulati. iTETRIS utilizza una organizzazione gerarchica per dedicare ad ogni modulo dell'architettura un proprio file di configurazione separato.

Una spiegazione completa di ogni parametro di configurazione messo a disposizione da iTETRIS esula dallo scopo di questa tesi ed è possibile trovarla in [Itb14]. Nel seguito saranno illustrati i parametri di utilizzati durante la valutazione del sistema e la raccolta dei risultati.

Configurazione di iCS

Questo file di configurazione è quello generale per impostare la simulazione, contiene infatti, oltre ai valori utilizzati per definire il comportamento del modulo iCS, quali file utilizzare per gli altri moduli.

In questo file sono specificati l'istante di inizio e di fine della simulazione, il rate di penetrazione della tecnologia utilizzata, la risoluzione utilizzata per il timestep della simulazione, la lista dei dispositivi infrastrutturali da simulare con la relativa posizione ed una serie di parametri dipendenti dal particolare scenario simulato.

Il valore del rate della penetrazione è stato mantenuto sempre costante al 100% in ogni simulazione eseguita, in quando è stato deciso di modificare quanti veicoli partecipassero effettivamente alla raccolta di dati all'interno dell'applicazione. Questo ha permesso due vantaggi: è possibile definire con precisione la percentuale delle varie classi di veicoli utilizzati ed ha permesso la raccolta di statistiche complete anche nel caso la penetrazione totale fosse inferiore al 100%. Se venisse impostato il valore generale presente in questo file i veicoli senza la tecnologia installata sarebbero scartati completamente da iCS e non sarebbe quindi possibile avere il valore reale con cui poter confrontare le prestazioni del sistema.

L'impostazione del valore della risoluzione non era presente all'interno della versione originale di iTETRIS ed è stata introdotta nel corso di questo progetto fra le modifiche effettuate al framework, come descritto nel paragrafo 5.2. La risoluzione originale di un secondo è infatti assolutamente inadeguata per la raccolta di risultati soddisfacenti nei protocolli utilizzati all'interno di questa tesi.

Questo file di configurazione contiene anche l'indicazione degli eseguibili per istanziare gli altri componenti del sistema ed i relativi file di configurazione. iCS si occupa infatti anche dell'avvio e della configurazione dei due simulatori e delle applicazioni clienti installate centralizzando le operazioni necessarie per l'avvio di una simulazione. Per avviare una simulazione occorre istanziare via riga di comando iCS specificando il solo percorso di questo file di configurazione generale.

Configurazione di SUMO

Per impostare correttamente SUMO occorre indicare gli specifici file che compongono lo scenario utilizzato; questi descrivono il reticolo stradale, la logica semaforica utilizzata nei vari incroci e contengono un elenco di veicoli ed il relativo percorso seguito durante la simulazione. Attraverso questi dati viene simulato lo stato di ogni veicolo ed è possibile ottenere in ogni momento le informazioni sulla sua posizione e la sua velocità istantanea.

Configurazione di NS-3

La configurazione di NS-3 consente di impostare correttamente sia le varie estensioni apportate durante l'integrazione con iCS sia i componenti dell'architettura

standard del simulatore. Si possono infatti selezionare differenti tecnologie di trasmissione wireless da installare all'interno dei nodi simulati ed impostarle opportunamente per il particolare scenario che si vuole eseguire.

Per quanto riguarda la configurazione dei numerosissimi parametri offerti si è scelto per semplicità di adottare una scelta analoga a quella fornita di default dalla piattaforma iTETRIS, ovvero di utilizzare i valori predefiniti dallo standard 802.11b in modalità ad-hoc, con metodologia di trasmissione DSSS [Tan02] ad 1 Mbps, ritardo di propagazione costante e modello logaritmico per la caduta del segnale. Come conseguenza delle precedenti impostazioni i nodi hanno una portata approssimabile a 170 metri. Questo valore è utilizzato per le RSU ed i veicoli di classe full, mentre per i nodi di classe media viene limitata la portata massima del segnale a 100m all'interno di iCS attraverso l'opportuno parametro nella sottoscrizione dedicata all'invio dei messaggi. Questa semplificazione si è resa necessaria perché non si è voluto modificare implementazione standard del sistema che prevede una sola classe di veicoli mobili. Questa approssimazione ha come effetto collaterale quello di sovrastimare il numero di collisioni dovute ai dispositivi medi in quanto viene utilizzato lo stesso raggio di propagazione degli altri veicoli all'interno di NS-3.

Configurazione dell'applicazione

L'architettura del simulatore iTETRIS lascia completa libertà allo sviluppatore nella creazione di applicazioni. In questo progetto si è scelto di creare una applicazione il più configurabile possibile ed è stato previsto l'utilizzo di un file in formato `xml` che racchiude le impostazioni sia relative alla parte di interfaccia con iCS sia quella dedicata all'implementazione dei vari protocolli presi in esame. È possibile impostare anche il file di output nel quale sono salvati i risultati della simulazione. Questo file di configurazione è passato all'applicazione quando viene invocata da iCS, che provvede ad interpretarlo e ad applicare le impostazioni prima di collegarsi al simulatore e di mettersi in attesa di comandi.

Un elenco dettagliato dei vari parametri presenti e dei valori utilizzati durante le simulazione è presente nell'Appendice A: per ogni attributo di ogni classe è presente un tag corrispondente all'interno di questo file, premettendo una personalizzazione completa del comportamento del protocollo.

Per automatizzare la raccolta dei risultati durante le simulazioni è stata prevista la lettura di un ulteriore file di testo dopo il caricamento delle impostazioni, contenente una limitata lista di parametri da modificare rispetto a quelli presenti nel file di confi-

gurazione. Questo meccanismo viene utilizzato da uno script per eseguire automaticamente numerose simulazioni automaticamente. I parametri che è possibile impostare sono le impostazioni del generatore di numeri casuali, la penetrazione dei veicoli full, medium e shadow e la modalità di formazione dei gruppi nel protocollo decentralizzato.

5.4 Risultati ottenuti dalle simulazioni

Per verificare il corretto funzionamento del sistema occorre anzitutto accertarsi che esso sia in grado di fornire una stima apprezzabile della quantità di traffico proveniente dalle diverse direzioni monitorate. In particolare i dati forniti alla RSU in esame devono essere tali da permettere di risalire alle condizioni reali del traffico anche attraverso l'uso di dati diversi dalla pura quantità di veicoli presenti in una data direzione, ad esempio attraverso la loro velocità o il tempo impiegato a superare l'incrocio. Come è stato mostrato nel capitolo 4 i due protocolli realizzati sono in grado di produrre anche altre informazioni sul traffico, tuttavia nel corso dei numerosi test effettuati e presentati in questo capitolo si è scelto di dare maggiore risalto ai dati riguardanti il numero di nodi rilevati e la loro velocità media essendo questi di maggiore criticità al fine di produrre una corretta comprensione della condizione del traffico che è necessario smaltire.

I test esposti nel seguito del presente capitolo si riferiscono ai risultati ottenuti utilizzando le configurazioni e gli scenari illustrati nel paragrafo 5.1. Dato che il sistema ricorre di frequente all'estrazione di numeri pseudocasuali all'interno delle proprie logiche di protocollo è necessario predisporre per ogni diversa tipologia di test l'esecuzione di più iterazioni, utilizzando gli stessi parametri ma un seme differente⁷ e procedendo in secondo luogo all'integrazione dei diversi risultati.

Ciascuno dei risultati di seguito esposti rappresenta quindi l'esito di un insieme di 10 iterazioni, ognuna della durata di 30 minuti⁸. Nel presentare i grafici riassuntivi si sceglie normalmente di escludere i primi 200 secondi di simulazione per presentare esclusivamente il comportamento del sistema in condizioni stabili, in quanto i dati presenti nelle due tracce utilizzate entrano a regime dopo alcuni minuti.

⁷ Il seme differente per il generatore di numeri pseudocasuali è condivisa sia dall'applicazione che da NS-3, mentre SUMO utilizza un valore costante per prevenire cambiamenti nella mobilità dei veicoli che renderebbe difficile il confronto tra le diverse esecuzioni di uno stesso scenario.

⁸ In seguito, se non esplicitamente indicato, tutti i valori temporali si riferiscono al tempo di sistema simulato.

I test svolti si focalizzeranno principalmente a mettere a confronto le prestazioni ottenute dal protocollo decentralizzato nelle due differenti implementazioni, quelle dei due differenti protocolli ed a confrontare diversi indicatori sul traffico nei casi di basse penetrazioni. Occorre infatti studiare e verificare se il ricorso al protocollo centralizzato comporta effettivamente un vantaggio in termini di qualità dei dati prodotti e capire eventualmente in quali circostanze esso si dimostra più vantaggioso rispetto a quello decentralizzato.

5.4.1 Indicatori di prestazione

Per poter valutare correttamente le prestazioni della soluzione realizzata e svolgere i confronti tra diversi scenari e protocolli è indispensabile disporre non solo dei risultati emessi dal sistema ma anche dei valori reali rispetto ai quali si sta tentando di ricavare una stima. Nel valutare i risultati relativi al numero di veicoli percepiti si confronteranno, ad esempio, i valori ottenuti dalla RSU di riferimento con il reale numero di veicoli presenti in quella direzione. I valori reali vengono estratti da una procedura detta *real mode* che si occupa di campionare e registrare in ogni secondo tutti i dati di interesse per l'analisi in corso. Tale routine dispone ovviamente di una completa conoscenza dell'ambiente simulato e dello stato completo dei suoi partecipanti ma non influisce in alcun modo con il comportamento dell'architettura in esame.

La procedura *real mode* è programmata per monitorare lo stato di tutti i veicoli che si trovano entro un raggio compreso tra 20 e 170 metri dalla RSU selezionata. Sotto i 20 metri i veicoli si avvicinano infatti al centro dell'incrocio e può risultare difficile distinguere l'esatta direzione a cui essi fanno riferimento; si ricorda inoltre che entrambi i protocolli sono impostati per terminare la loro attività superata tale distanza dalla RSU. La scelta di non tracciare i veicoli a distanze superiori a 170 metri dall'incrocio riflette il range massimo ottenibile dai veicoli di classe più elevata: oltre tale soglia essi non possono in ogni caso comunicare con la RSU e non sarebbe quindi corretto includerli all'interno del conteggio.

I valori ottenuti mediante la modalità reale sono da considerarsi ideali soltanto in relazione al range monitorato e non in maniera assoluta: nonostante i nodi non riescano effettivamente a comunicare con la RSU oltre il limite preimpostato, nel protocollo decentralizzato questi possono comunque fornire, una volta entrati nel range operativo, dei dati che comprendono il contributo di veicoli a distanza leggermente superiore alla soglia dei 170 metri. In simili casi la differenza tra i valori ottenuti in

real mode e quelli ricavati dai protocolli è da intendersi a vantaggio del sistema, considerato che quest'ultimo riesce a coprire una distanza superiore al campo visivo della modalità reale.

Per contestualizzare meglio i grafici successivi si è scelto di riportare, ove appropriato, una curva che riporta la distanza del veicolo più lontano rilevato in ogni secondo dalla RSU di riferimento utilizzando le informazioni ricevute dal protocollo. Confrontando tale curva con il range costante usato nella modalità reale è possibile dare un'interpretazione più significativa alle differenze tra i valori ideali e quelli effettivamente ottenuti.

Nello studiare e commentare il comportamento dell'architettura implementata, si vorrebbe poter estrarre da ogni configurazione simulata un indice di performance che determini approssimativamente il livello di qualità dei dati raggiunto dal sistema rispetto ad una particolare configurazione presa come riferimento. In generale questo requisito non è semplice da soddisfare, considerando la quantità e l'eterogeneità delle informazioni prodotte e tenendo conto che è quasi impossibile disporre di valori ottimi con cui poterle confrontare.

Per rispondere almeno in parte a questa esigenza è stato introdotto un indicatore che rappresenta il rapporto in ogni secondo tra il valore della grandezza scelta di una particolare configurazione e quello ottenuto nella simulazione presa come riferimento. Questo valore è espresso in percentuale ed indica di quanto si discostano le due simulazioni per la particolare grandezza scelta. Per meglio valutare le informazioni ottenute si calcolerà questo indicatore per diverse tipologie, come ad esempio il numero di veicoli rilevati per una particolare direzione o la loro velocità media. Per sintetizzare in un unico valore la performance di una simulazione è introdotta una media dei dati calcolati, denominata delta (Δ).

La possibilità di paragone di differenti informazioni ottenute in due simulazioni sarà molto utile per confrontare il comportamento del sistema con penetrazioni della tecnologia variabile in quanto permette di capire come reagiscono le grandezze scelte a questi cambiamenti.

I primi risultati sperimentali che verranno presentati riguardano una verifica del corretto funzionamento del protocollo all'interno del simulatore iTETRIS attraverso un confronto con i risultati ottenuti nella versione originale di NS-3; si cercherà poi di confrontare i due protocolli discussi a fronte di cambiamenti della composizione della popolazione al variare delle tre classi di veicoli enunciate nel paragrafo 2.4.

Verrà infine condotto uno studio per comprendere come il fattore di penetrazione della tecnologia proposta incide sulla qualità dei dati raccolti dal sistema.

5.4.2 Confronto fra le implementazioni

Il primo test presentato riguarda una verifica dell'implementazione del sistema all'interno del simulatore iTETRIS per confrontarne le prestazioni con quelle ottenute nella versione standard di NS-3. Le figure 5.3 e 5.4 mostrano i risultati ottenuti rispettivamente nello scenario Andrea Costa e Pasubio nella condizione ideale di una penetrazione totale della tecnologia di tipo full. Ciascuna delle due figure rappresenta l'andamento dei valori in due direzioni di interesse sia per la procedura di formazione reattiva che per quella proattiva: le linee relative alle performance dei protocolli ottenute con iTETRIS sono mostrate sul grafico in rosso, mentre quelle relative ad NS-3 sono in blu. Le linee scure si riferiscono sempre ai valori ottenuti dalla modalità reale.

Ad un primo impatto si nota che le due implementazioni tendono in generale ad avere andamenti non perfettamente identici, specialmente nello scenario Pasubio, nel quale si ha una prestazione della versione di NS-3 migliore. Questa differenza è chiaramente del tutto prevedibile data la grande diversità nell'architettura dei due simulatori ed è dovuta a differenti fattori. In primo luogo occorre ricordare che la gestione degli eventi è notevolmente diversa nelle due piattaforme: come illustrato nel paragrafo 3.1, l'esecuzione di una simulazione all'interno di iTETRIS è suddivisa in step temporali costanti, del valore di 25ms nel caso del protocollo decentralizzato, per cui la reazione agli eventi dei vari componenti del protocollo avviene necessariamente in istanti temporali non precisi e con un ritardo che dipende da questo intervallo di tempo.

Se ad esempio un nodo ricevesse un messaggio dopo 10ms dall'inizio di uno step di simulazione, prima di essere informato di questo evento e poter eseguire un'opportuna risposta deve attendere la fine del ciclo, per cui reagirà con circa 15ms di ritardo. Queste limitazioni ed il loro possibile impatto sulle logiche dei protocolli non sono ovviamente presenti all'interno della versione originale implementata all'interno di NS-3, nella quale i componenti sono attivati quando necessario senza alcun tipo di ritardo. In secondo luogo, come illustrato nel paragrafo 5.2, la risoluzione utilizzata non è quella che permette di ottenere i migliori risultati in assoluto, ma è stato necessario scendere a compromessi fra le prestazioni ottenute dal protocollo ed il tempo fisicamente necessario per eseguire le simulazioni.

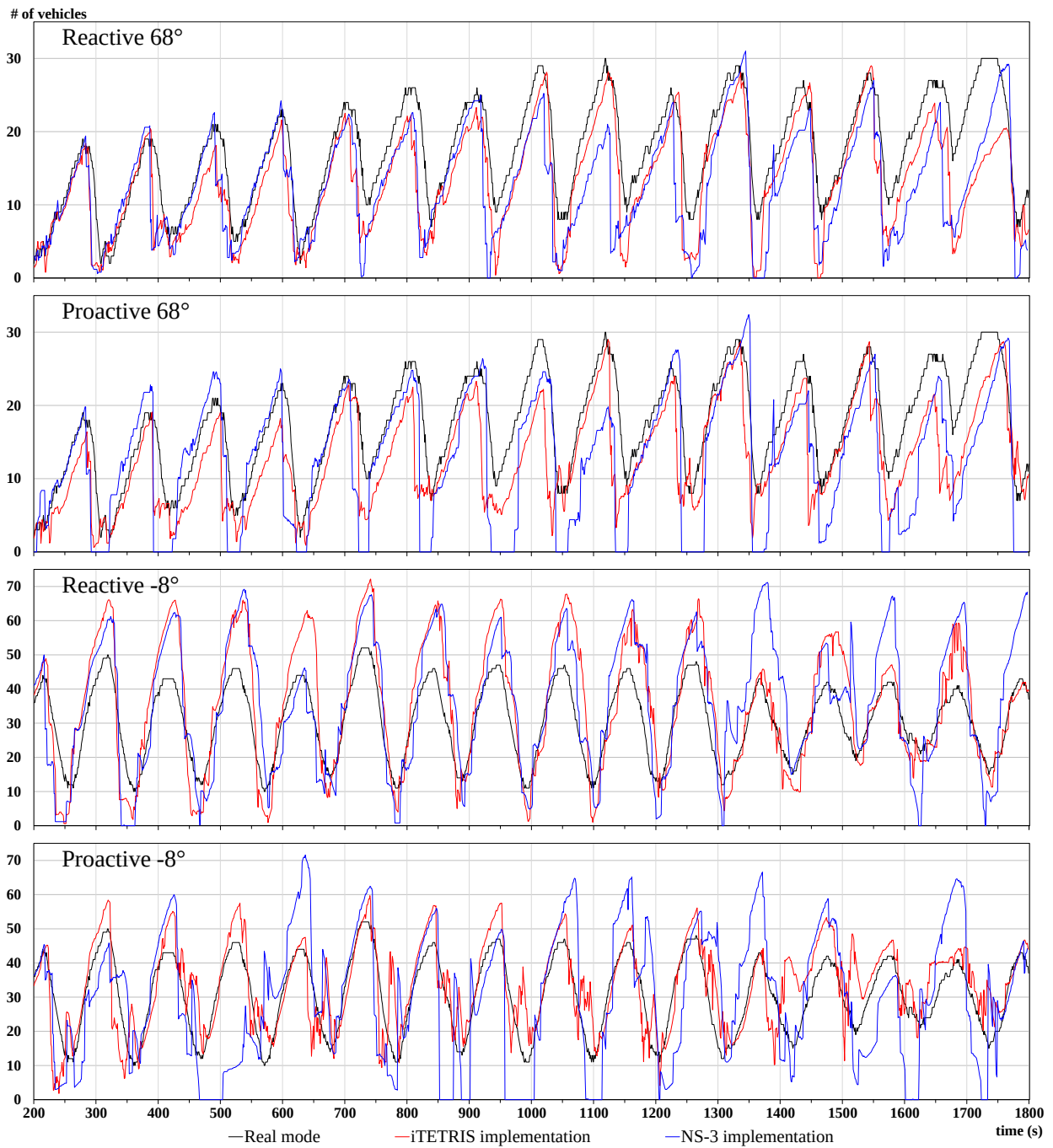


Figura 5.3: Confronto dell'implementazione del protocollo decentralizzato in NS-3 ed iTETRIS per lo scenario A. Costa nelle direzioni 68° e -8° . I grafici indicano il numero di veicoli rilevati da ciascuna implementazione e quelli ottenuti con la modalità reale. Il primo grafico per ogni direzione si riferisce alla modalità di formazione dei gruppi reattiva, il secondo a quella proattiva.

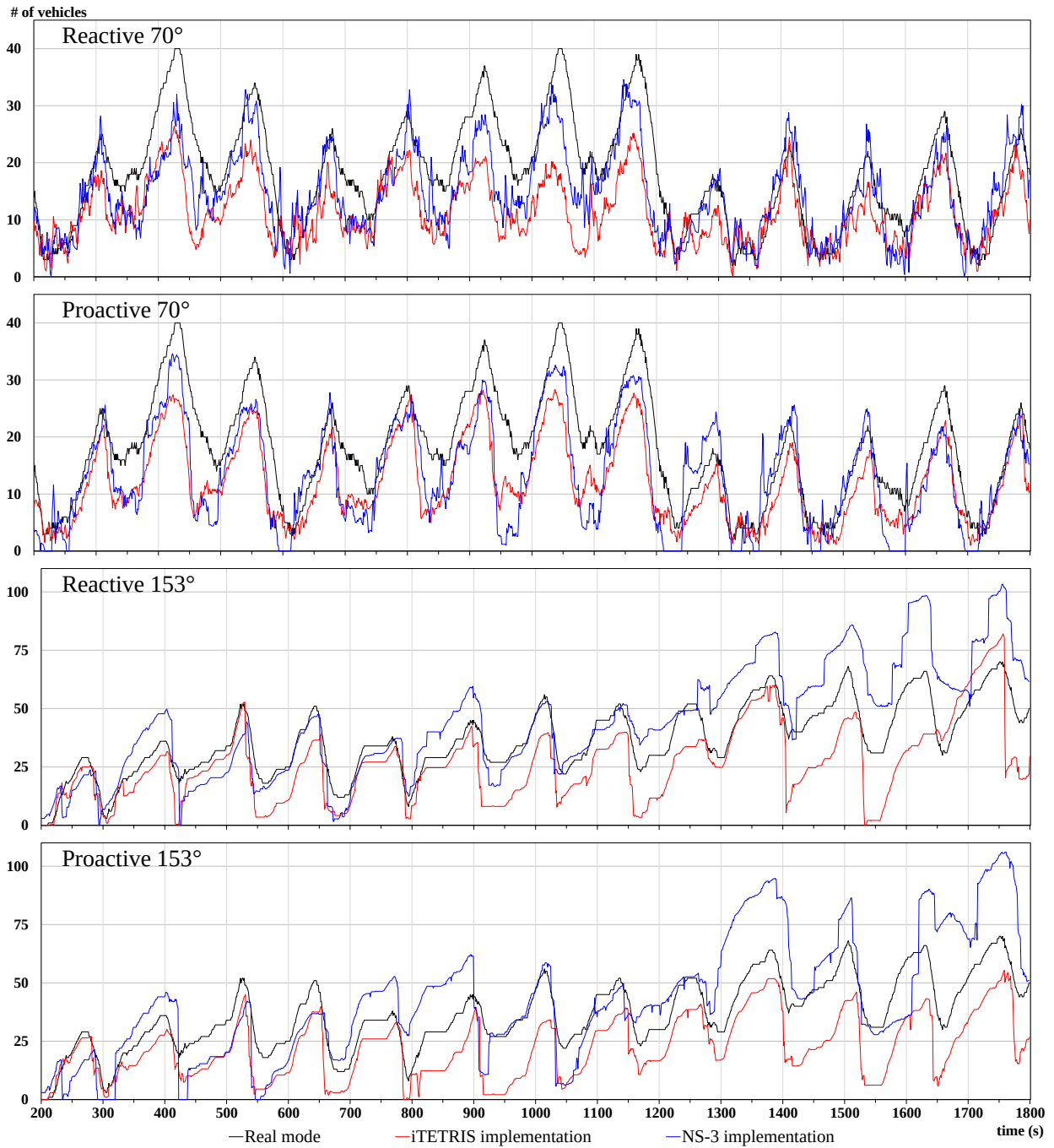


Figura 5.4: Confronto dell'implementazione del protocollo decentralizzato in NS-3 ed iTETRIS per lo scenario Pasubio per le direzioni 70° e 153°. I grafici indicano il numero di veicoli rilevati da ciascuna implementazione e quelli ottenuti con la modalità reale. Il primo grafico per ogni direzione si riferisce alla modalità di formazione dei gruppi reattiva, il secondo a quella proattiva.

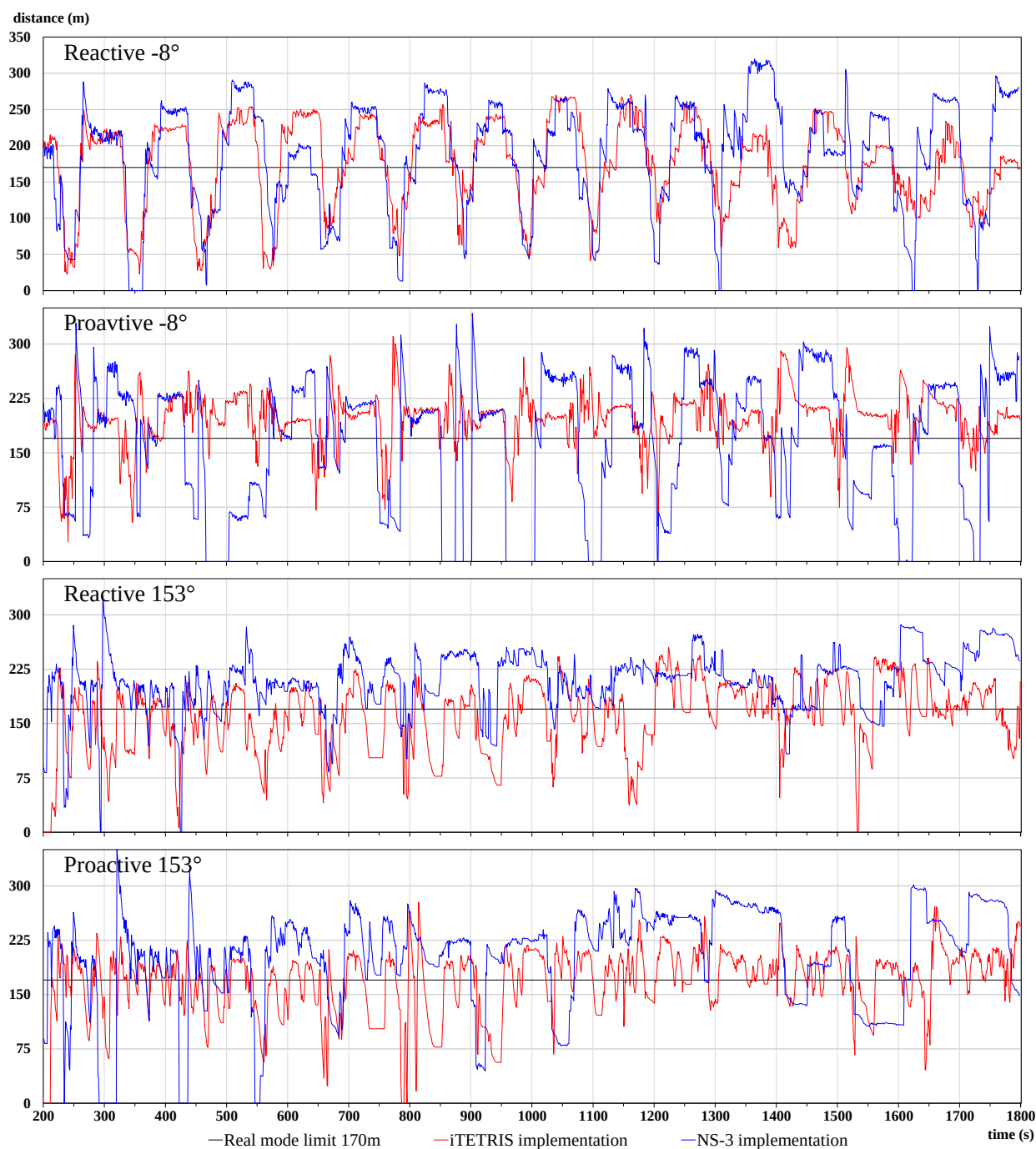


Figura 5.5: Confronto dell'implementazione del protocollo decentralizzato in NS-3 ed iTETRIS per lo scenario A. Costa nella direzione -8 e Pasubio nella direzione 153°. I grafici mostrano la distanza del veicolo più lontano rilevata dal protocollo per ciascuna implementazione ed il limite della modalità reale di 170 metri. Il primo grafico per ogni direzione si riferisce alla modalità di formazione dei gruppi reattiva, il secondo a quella proattiva.

Vi possono essere anche altri fattori minori che possono influenzare le prestazioni ottenute, come la diversa versione del simulatore NS-3 utilizzata all'interno di iTETRIS rispetto a quella adoperata per la raccolta dei dati nell'implementazione originaria.

Inoltre nelle diverse simulazioni è stata mantenuta la stessa configurazione dei componenti che compongono il protocollo, per ottenere dei risultati che fossero il più possibile confrontabili tra di loro. È evidente come la differente natura della piattaforma iTETRIS possa beneficiare da vincoli temporali più laschi nelle varie fasi del protocollo in modo tale da mitigare il ritardo introdotto. Occorre però prestare particolare attenzione nel cambiamento di questi parametri in quanto si potrebbe modificare significativamente il comportamento del sistema, rendendo difficile il confronto tra i differenti risultati ottenuti.

Confrontando i tracciati ottenuti dai protocolli con il reale numero di vetture presenti si può osservare in alcuni casi una sottostima della reale quantità di nodi presenti nel tratto stradale, imputabile a diversi fattori. Il più importante tra questi riguarda il fatto che l'attività di un gruppo viene sempre terminata nel momento in cui il leader raggiunge una distanza di 20 metri dalla RSU. Superato tale limite si ha un immediato crollo del numero di nodi rilevati dal sistema in quanto potrebbero venire disattivati anche dei veicoli che si trovano ancora relativamente lontani dall'incrocio (e quindi ancora rilevati nella modalità reale).

Un altro fattore riguarda il rischio sempre presente che alcuni nodi escano anche solo temporaneamente da un gruppo a causa di un valore di direzione non conforme a quello previsto dalla RSU oppure per via di alcune iterazioni di probe fallite a seguito di alcune perdite di pacchetti dovute ad un elevato numero di collisioni.

Questi difetti possono in generale essere in parte corretti attraverso degli interventi al sistema finalizzati a perfezionare la gestione del ciclo di vita dei singoli gruppi ma possono anche essere sensibilmente attenuati da una logica aggiuntiva collocata internamente alle varie RSU. Compito di tale ipotetico modulo dovrebbe essere quello di raffinare i dati ottenuti direttamente dai singoli gruppi, applicando un filtro per eliminare variazioni improvvise o errori e per migliorare le informazioni disponibili a seconda del loro contesto e soprattutto avendo a disposizione lo stato corrente dei segnali di semaforici emessi dalla RSU.

Si consideri ad esempio il caso in cui viene rilevata la chiusura di un gruppo dotato di una velocità media particolarmente bassa contestualmente all'emissione di

un segnale di stop per quella direzione da parte della RSU. Una logica sufficientemente intelligente potrebbe facilmente intuire che in realtà i veicoli appartenenti al gruppo appena chiuso non possono avere oltrepassato l'incrocio, visto che la loro velocità era prossima allo zero ed il semaforo stava segnalando loro di fermarsi. Viceversa, se la chiusura di un gruppo viene riscontrata quando le vetture hanno accesso all'incrocio è possibile rinforzare ancora per qualche secondo la loro presenza basandosi sulla velocità rilevata e sul tratto stradale che ancora devono percorrere prima di raggiungere la RSU.

Procedendo più concretamente all'analisi dei risultati si osserva innanzitutto l'andamento sinusoidale del numero dei veicoli presenti nelle diverse direzioni. Questo è chiaramente dovuto al fatto che in condizioni di traffico intenso i veicoli procedono dapprima ad accodarsi in attesa di poter accedere all'incrocio ed in seguito superano la RSU abbandonano il tratto stradale monitorato non appena ricevono un segnale verde dal semaforo. Nel complesso i protocolli riescono a seguire piuttosto fedelmente il reale andamento del traffico, pur presentando sottostime e, meno frequentemente nella versione implementata in iTETRIS, sovrastime.

Le direzioni che raggiungono valori di densità maggiore negli scenari di A. Costa e Pasubio sono rispettivamente quelle di -8° e 153° ; incidentalmente tali tratti stradali dispongono anche di un maggior numero di corsie per senso di marcia (rispettivamente 3 e 4). Nel caso della direzione -8° si verificano frequenti picchi che eccedono il numero di veicoli rilevati in modalità reale. In realtà, come mostrato in Figura 5.5, tale sovrastima non è dovuta a degli errori del sistema ma è imputabile alle lunghe code che si presentano nel tratto stradale in esame: al crescere del numero di veicoli che si accodano in attesa di accedere all'incrocio, la RSU riceve frequentemente dei riscontri da gruppi a distanza elevata i cui membri superano molto spesso i 200 metri di distanza, fornendo pertanto dei dati qualitativamente migliori rispetto a quelli ideali ma limitati a una visione di 170 metri.

Nel caso dell'implementazione in iTETRIS le imprecisioni riscontrate nel tratto di direzione 153° dello scenario Pasubio sono invece sintomo dell'inaccuratezza del protocollo in condizioni di densità estremamente elevate, nonostante anche in questo caso la Figura 5.5 mostri la capacità del sistema di lavorare su un'area più ampia rispetto a quella impostata nella modalità reale. Le imprecisioni non sono tali da compromettere in maniera critica il funzionamento complessivo del sistema ma in alcuni casi, specialmente quando la densità raggiunge valori di circa 70 veicoli, si assiste ad una sottostima abbastanza rilevante, superiore anche alle 15 unità. Il fatto che tale

tratto stradale sia oltretutto quello che dispone del più elevato numero di corsie (quattro) tra tutte le simulazioni suggerisce che l'elevato numero di veicoli che si trovano a comunicare in un'area relativamente ristretta comporta necessariamente delle difficoltà per il sistema. Si può comunque ricavare un elemento positivo tenendo conto che nelle situazioni di elevata densità la modalità di formazione proattiva presenta una precisione leggermente migliore tra i due protocolli.

La versione NS-3 del protocollo è meno affetta da queste limitazioni e riesce spesso ad ottenere risultati migliori rispetto alla modalità reale in questa direzione, soprattutto nella seconda parte della simulazione dove trae beneficio dai dati ricevuti da veicoli che sono a distanze fino a circa 300 metri dalla RSU. Si ha però un maggior rumore, soprattutto nella modalità di formazione proattiva, nella quale in alcuni momenti è difficile capire l'andamento del flusso del traffico. Nonostante questo la maggiore reattività garantita dall'architettura di questo simulatore porta inevitabilmente dei benefici in questo scenario caratterizzato da una densità di veicoli molto alta.

Osservando le differenti implementazioni si può inoltre notare come le distanze massime ottenute in NS-3, in blu, siano quasi sempre superiori rispetto a quelle in iTETRIS, mostrate in rosso. Questa discrepanza motiva in parte le minori prestazioni di quest'ultima versione ed è causata molto probabilmente dal ritardo introdotto dalla piattaforma che impedisce una reazione veloce dei nodi ai messaggi ricevuti.

Soprattutto nello scenario di A. Costa, si può notare come l'implementazione in iTETRIS presenti un risultato generalmente più stabile e con meno rumore, sia per quando riguarda il numero di veicoli rilevati, che per il valore di distanza dei nodi che partecipano alla raccolta dei dati. Questo comportamento è particolarmente accentuato per nella modalità proattiva di formazione dei gruppi e per la direzione 68°.

Volendo quantificare la differenza fra le prestazioni delle due versioni si è proceduto a calcolare il valore del delta dai risultati sul numero di vetture rilevate ottenuti, prendendo come riferimento quelli estratti dall'implementazione in NS-3, come mostrato in Figura 5.6. Nel grafico è mostrata la differenza media percentuale fra i risultati ottenuti nei due casi per ogni direzione monitorata all'interno dei due scenari. Si può notare un comportamento mediamente peggiore rispetto al riferimento, anche se i risultati acquisiti sono abbastanza variegati. Nel caso della direzione 70° di Pasubio si ha ad esempio una prestazione media peggiore per la modalità di formazione reattiva e migliore per quella proattiva; nella direzione -113° ed in parte anche in quella 170° di A. Costa il risultato è invece opposto. Lo scenario di A. Costa è indubbiamente più favorevole all'implementazione realizzata rispetto a quello di Pasubio.

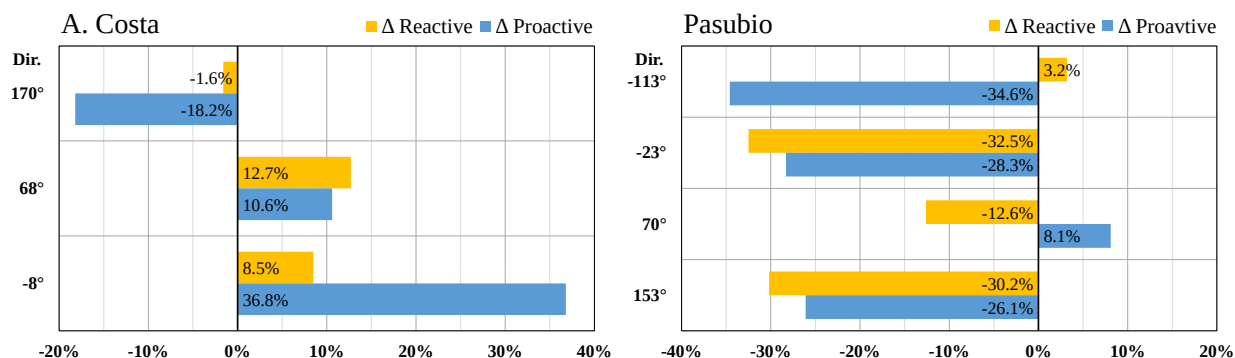


Figura 5.6: Valore delta calcolato sul numero di vetture rilevate. I grafici mostrano le prestazioni dell'implementazione in iTETRIS rispetto a quella in NS-3 per entrambi gli scenari utilizzati. Sono indicati i valori per ogni direzione sia nel caso di modalità di formazione reattiva che proattiva.

Benché queste differenze siano comunque non trascurabili, si possono ritenere le due implementazioni abbastanza simili considerando soprattutto le funzionalità aggiuntive offerte da iTETRIS, soprattutto per quanto riguarda la verifica delle euristiche di gestione del traffico che potranno utilizzare i dati raccolti in tempo reale. Si è pertanto ritenuto accettabile l'implementazione ottenuta che, seppure non perfetta, permette comunque di ottenere risultati comparabili ed in ogni caso decisamente migliori rispetto a quelli possibili con la versione standard della piattaforma iTETRIS, come mostrato in Figura 5.2.

Nel seguito di questo capitolo tutti i risultati mostrati faranno riferimento all'implementazione dei protocolli all'interno della piattaforma di simulazione iTETRIS.

5.4.3 Confronto fra i protocolli

Come illustrato nel paragrafo 4.2 il protocollo centralizzato è stato sviluppato interamente in iTETRIS senza passare prima da altri simulatori, ed ha come obiettivo principale il miglioramento delle prestazioni ottenibili in scenari di basse penetrazioni delle tecnologie studiate.

Prima di passare alla verifica dei risultati in questi scenari occorre però confrontare i risultati di questo nuovo protocollo con quelli ottenuti dalla versione decentralizzata, per verificarne il corretto funzionamento e le capacità di sintesi di una stima accurata delle condizioni del traffico. In Figura 5.7 è mostrato un confronto fra le due versioni nelle direzioni 68° e -8° di A. Costa ed in quelle 70° e 153° di Pasubio: la linea scura rappresenta il numero di veicoli rilevati dalla modalità reale, quella verde i risultati del protocollo centralizzato, mentre quelle rosse e blu rappresentano i valori

ottenuti dal protocollo decentralizzato con modalità di formazione dei gruppi, rispettivamente reattiva e proattiva.

Osservando i grafici riportati in figura si può subito notare che i risultati ottenuti da questo nuovo protocollo centralizzato sono molto più vicini ai valori rilevati attraverso la modalità reale. Questo risultato potrebbe essere previsto considerando che, per come è stato progettato, il funzionamento è molto simile alla procedura utilizzata per ottenere il reale numero di veicoli presenti. Analizzando in dettaglio le tracce si osserva che i valori ottenuti sono molto più costanti rispetto a quelli del protocollo decentralizzato ed hanno un discostamento dalla traccia reale inferiore.

Si può notare una sovrastima generalmente contenuta che tende ad essere abbastanza costante in entrambi gli scenari, ad eccezione della traccia nella direzione 70° di Pasubio nella quale in alcuni punti il valore reale sono sottostimati; il valore offre comunque una approssimazione migliore del flusso del traffico rispetto a quella ottenuta dal protocollo decentralizzato in entrambe le sue modalità. Questa sovrastima può essere spiegata dal fatto che un veicolo, dopo aver superato la soglia dei 20 metri dalla RSU, invia un ultimo messaggio per indicare la sua uscita dalla raccolta dati e viene quindi conteggiato anche se la sua posizione è interna a quella stabilita, a differenza di quanto avviene nella procedura di raccolta dei dati reale la quale scarta immediatamente un nodo subito dopo il superamento di tale soglia.

Tornando a soffermarsi sulle direzioni maggiormente trafficate, -8° per lo scenario A. Costa e 153° per Pasubio, si può notare che per la prima vi è una quasi totale assenza di sovrastima dei dati reali se confrontato con i risultati ottenuti nel protocollo decentralizzato, dovuta all'impossibilità che questa versione ha di comunicare con veicoli a distanze superiori al raggio di copertura della RSU. Nonostante questa limitazione e la grande quantità di nodi contemporaneamente all'interno del raggio di copertura della RSU i dati raccolti per la direzione 153° sono apprezzabilmente migliori se confrontati con i risultati dell'altro protocollo.

Nei grafici della Figura 5.8 si può apprezzare la differenza nell'area di funzionamento dei due protocolli: mentre quello decentralizzato arriva a raccogliere dati di veicoli che sono in alcuni punti fino a più di 300 metri di distanza dall'incrocio, quello centralizzato non supera mai la distanza massima prevista come range di trasmissione dei veicoli, rimanendo abbastanza stabilmente intorno ai 150 ÷ 160 metri. Questa differenza non è inaspettata, deriva infatti dalla scelta progettuale di non utilizzare nessun tipo di strategia di routing per consentire la propagazione dei messaggi oltre l'area di copertura diretta delle trasmissioni.

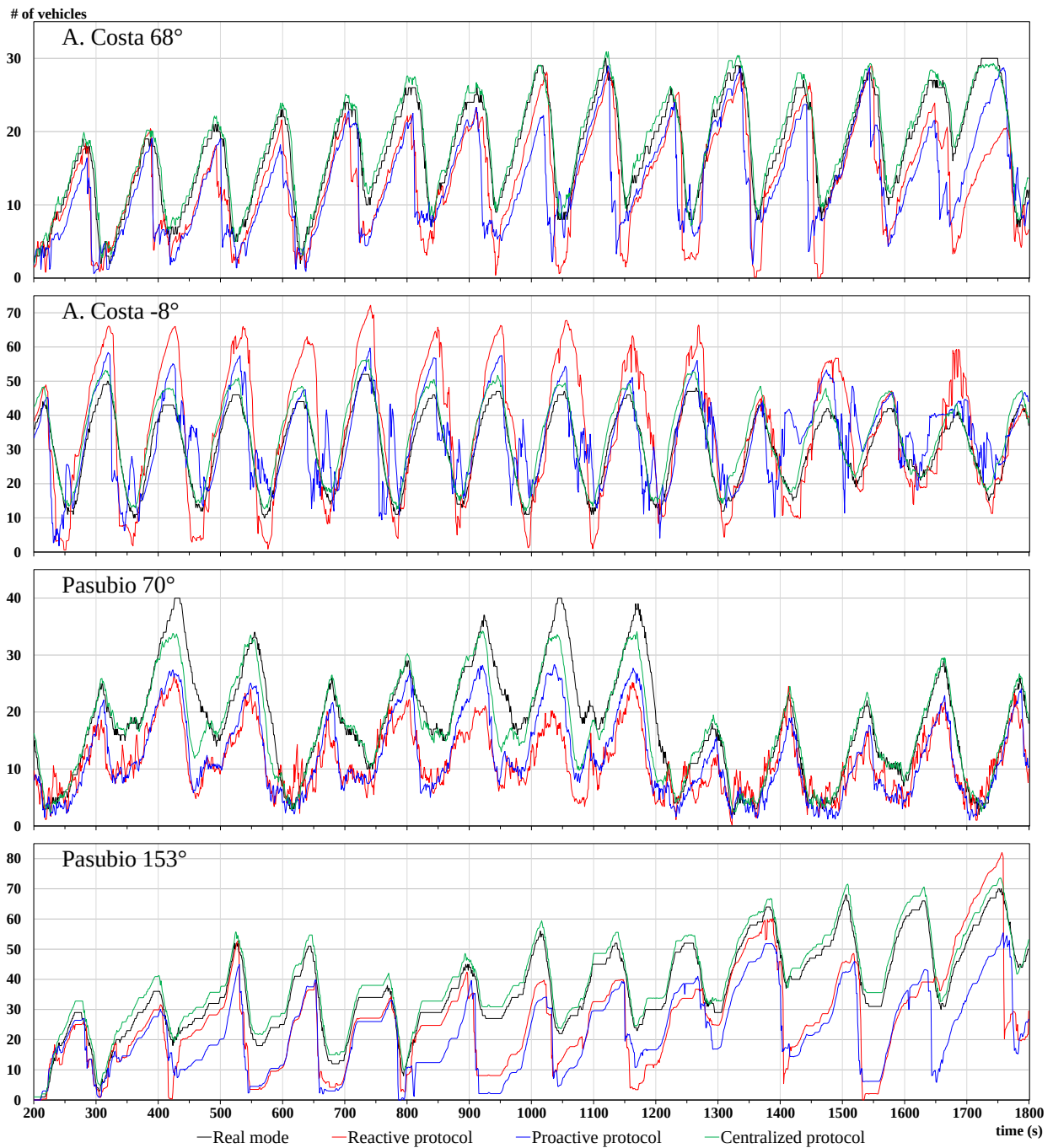


Figura 5.7: Confronto dei due protocolli per gli scenari di A. Costa e Pasubio per le direzioni 68°, -8°, 70° e 153°. I grafici indicano il numero di veicoli rilevati da ciascuna versione e quelli ottenuti attraverso la modalità reale. Nel protocollo decentralizzato sono presenti entrambe le modalità di formazione dei gruppi.

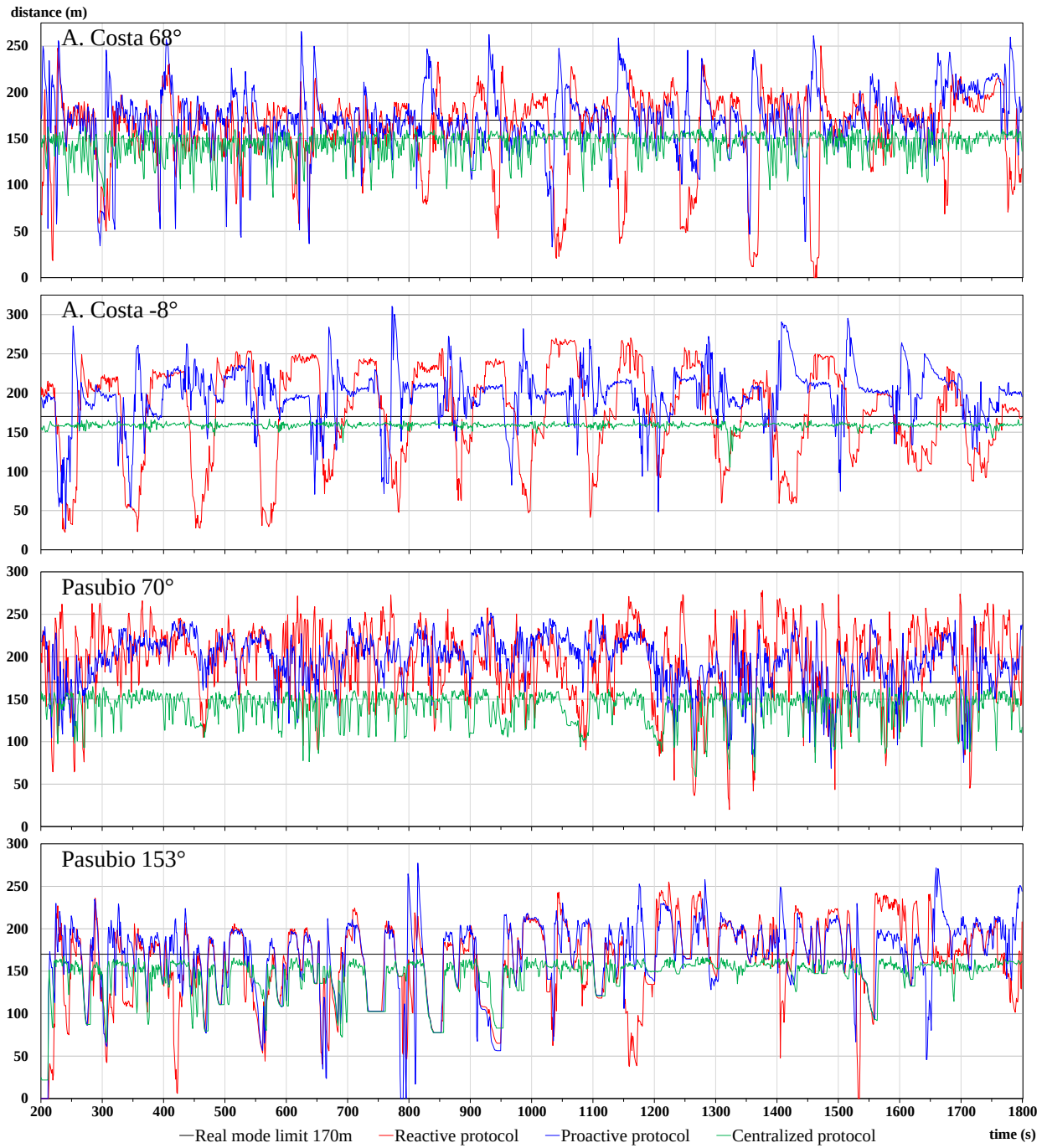


Figura 5.8: Confronto dei due protocolli per gli scenari di A. Costa e Pasubio per le direzioni 68°, -8°, 70° e 153°. I grafici mostrano la distanza del veicolo più lontano rilevata dal protocollo per ciascuna versione ed il limite della modalità reale di 170 metri. Nel protocollo decentralizzato sono presenti entrambe le modalità di formazione dei gruppi.

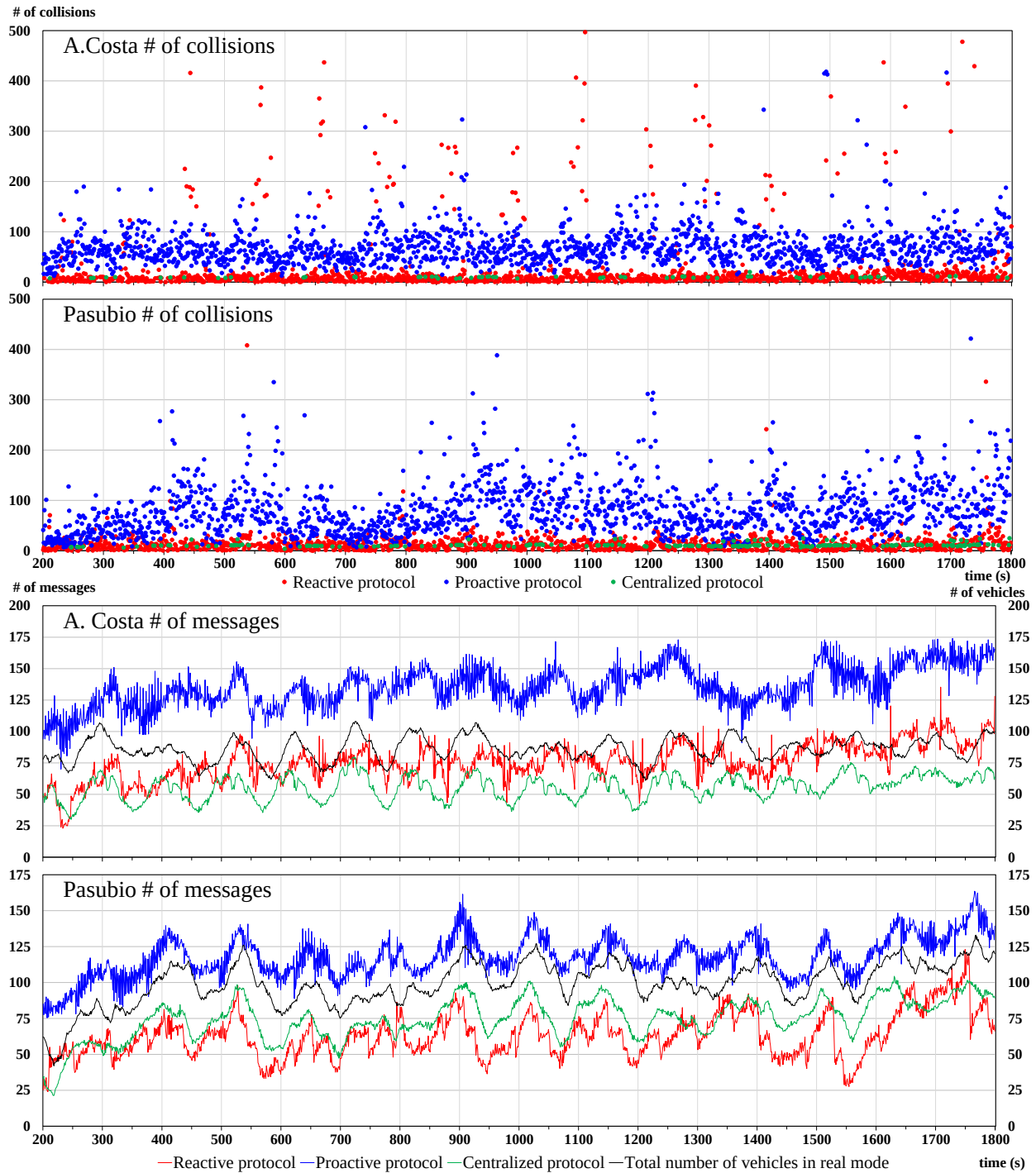


Figura 5.9: Confronto dei due protocolli per gli scenari di A. Costa e Pasubio. I primi due grafici mostrano il numero totale di nodi che soffrono una collisione in una loro trasmissione, gli altri due mostrano il numero totale di messaggi inviati dai nodi ed il numero di veicoli totale fornito come riferimento per contestualizzare i dati. Nel protocollo decentralizzato sono presentate entrambe le modalità di formazione dei gruppi.

Considerata questa limitazione i risultati ottenuti sono ancora più significativi: il protocollo riesce generalmente a rilevare tutti o quasi i veicoli presenti all'interno del suo range di funzionamento. Alla luce di questi dati le prestazioni ottenute dal protocollo decentralizzato appaiono carenti, soprattutto considerando che i dati ottenuti sono solitamente inferiori, nonostante partecipino alla raccolta delle informazioni anche veicoli molto più lontani della soglia di 170 metri stabilita come limite della modalità reale e come portata dei segnali trasmessi. Nel paragrafo 5.4.5 verranno effettuati ulteriori test con veicoli aventi range di trasmissione minore (la classe medium) per verificare come questa limitazione del protocollo centralizzato influisce sui risultati ottenuti e se quello decentralizzato offre un'alternativa migliore in questo caso.

Paragonando meglio le due diverse strategie di formazione dei gruppi nel protocollo decentralizzato si evidenziano comunque alcuni dubbi sull'effettiva utilità della modalità di tipo proattivo. Nelle direzioni a densità più bassa (A. Costa 68° e Pasubio 70°) il comportamento è pressoché identico mentre nei tratti più difficoltosi (A. Costa -8° e Pasubio 153°) i due andamenti si discostano frequentemente con quello della modalità proattiva più irregolare e con errori occasionali. Nella direzione -8° di A. Costa la modalità reattiva offre risultati migliori rispetto alle altre versioni sperimentate, in quanto i valori ottenuti sono frequentemente maggiori di quelli rilevati dalla modalità reale. Il confronto tra le due modalità di formazione verte a favore di quella di tipo reattivo a causa della maggiore stabilità presentata e, soprattutto, del costo inferiore in termini di messaggi inviati ed il minore rumore prodotto.

Queste considerazioni trovano conferma nei test condotti per verificare quanto i diversi protocolli costino in termini di messaggi inviati e di collisioni provocate. Nella Figura 5.9 è possibile osservare i risultati aggregati di tale studio per entrambi gli scenari utilizzati. Per entrambi gli scenari il grafico superiore mostra tre diverse nuvole di punti ciascuna rappresentativa del numero di nodi coinvolti ogni secondo in una collisione tra più trame di pacchetti (ricevuti o inviati); quello inferiore raffigura il numero di veicoli rilevati ogni secondo dalla modalità reale e le curve relative alla quantità di pacchetti inviati dai nodi a seconda del protocollo di utilizzato.

Dai grafici si evince facilmente che il ricorso alla modalità proattiva, che comporta l'utilizzo del modulo NodeMap, genera un maggior numero di pacchetti persi a causa del maggiore quantitativo di messaggi mediamente inviati da ogni nodo. Anche se la strategia proattiva è stata concepita per poter effettuare un'elezione più veloce e meno rumorosa in condizioni di elevata densità, il costo in termini di messaggi CAM

necessari al suo funzionamento supera decisamente questi vantaggi. È comunque doveroso ricordare che le potenzialità offerte dalla disponibilità di una mappatura dei nodi circostanti possono spingersi ben oltre al semplice protocollo di formazione del gruppo e costituiscono il fondamento di molte potenziali funzionalità che possono essere aggiunte all'architettura.

Soprattutto nello scenario di A. Costa si possono inoltre notare alcuni picchi nelle collisioni sofferte dai nodi nella modalità reattiva: questi sono dovuti alla grande concentrazione di messaggi scambiati dal processo di formazione dei gruppi, che possono andare a collidere con le trasmissioni di altri nodi impiegati nella raccolta di dati.

I dati relativi al protocollo centralizzato sono quasi invisibili nei grafici superiori, in quanto presentano una quantità di collisioni che è di circa un ordine di grandezza inferiore rispetto a quella del protocollo decentralizzato. I forti limiti imposti alle possibilità di comunicazione dei nodi, illustrati nel paragrafo 4.2.1, si sono rilevati molto utili per limitare le interferenze reciproche fra i veicoli e risultano in una modalità di comunicazione molto efficiente, che garantisce una raccolta dati ottimale. L'andamento del numero di messaggi scambiati da questo protocollo segue fedelmente la quantità di veicoli rilevati dalla modalità reale. Nonostante questo numero sia simile, o superiore nel caso di Pasubio, a quello della modalità reattiva del protocollo decentralizzato, le collisioni risultanti sono molto inferiori a testimonianza dell'utilità di una politica di coordinamento delle comunicazioni dei vari nodi. Il protocollo decentralizzato risulta carente sotto questo punto di vista: la totale mancanza di un'organizzazione delle trasmissioni dei nodi quando sono nella fase di creazione dei gruppi nella modalità reattiva, e fra differenti plotoni durante la raccolta dei dati, risulta in un alto numero di collisioni nonostante sia scambiato un numero paragonabile di messaggi.

Nonostante il protocollo centralizzato fosse stato progettato per offrire prestazioni migliori soprattutto in scenari di bassa penetrazione, i test effettuati mostrano che si comporta egregiamente anche nel caso di utilizzo in ambienti molto affollati. Occorre inoltre ricordare che nei test viene utilizzata una temporizzazione statica per stabilire l'intervallo di risposta ai beacon concesso ai nodi, che rimane costante sia in funzione della direzione che delle condizioni del traffico. Adottare una politica dinamica potrebbe portare ad una ulteriore diminuzione delle interferenze reciproche sofferte dai nodi.

Per terminare questo primo confronto fra i vari protocolli sono proposti i grafici dell'indicatore delta sul numero di nodi rilevati, prendendo come riferimento i dati

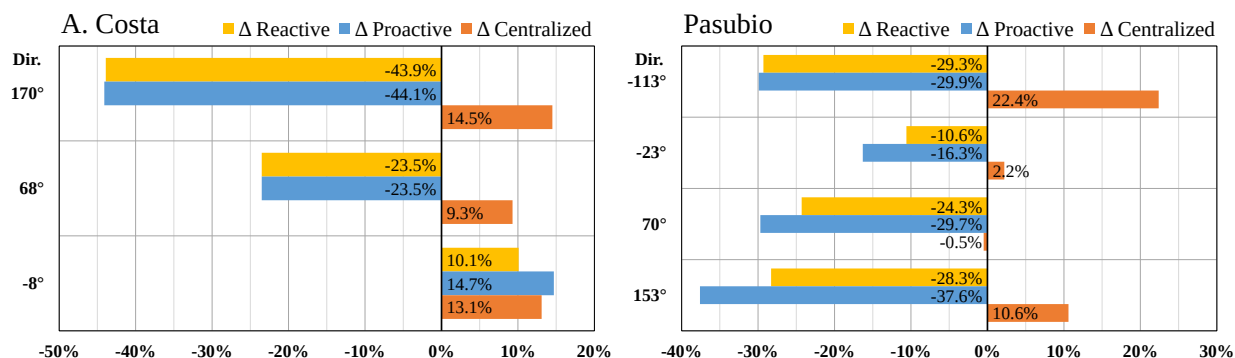


Figura 5.10: Valore delta calcolato sul numero di vetture per il protocollo centralizzato e le due modalità di quello decentralizzato rispetto ai valori reali ottenuti nella modalità reale. I grafici mostrano le prestazioni per ogni direzione dei due scenari utilizzati.

rilevati dalla modalità reale. Come si può vedere nella figura 5.10, il numero di veicoli rilevati dal protocollo centralizzato è, tranne per la direzione 70° di Pasubio, sempre maggiore rispetto a quelli ottenuti dalla modalità reale. Passando alla valutazione del protocollo decentralizzato si evidenziano evidenti carenze per ogni direzione, con l'unica eccezione della direzione -8° dello scenario A. Costa. La modalità proattiva inoltre funziona generalmente peggio rispetto a quella reattiva, a causa dell'alto numero di trasmissioni e di collisioni che la caratterizzano. Una diminuzione della frequenza dei messaggi di heartbeat, impostata a 2 secondi per questi test, potrebbe portare ad ottenere prestazioni migliori.

5.4.4 Analisi delle informazioni sul traffico raccolte

Nel paragrafo precedente sono stati analizzati i risultati riguardanti il numero di nodi rilevati da entrambi i protocolli, nel seguito saranno presentati ulteriori test per capire se esistono altre informazioni utilizzabili dalle singole RSU per ottenere una migliore conoscenza sull'andamento del flusso del traffico dei veicoli in una direzione. Verranno infatti analizzati i valori sulla velocità dei nodi e sul tempo da loro impiegato per attraversare l'incrocio.

Il parametro relativo al numero di nodi rilevati dai protocolli è stato già commentato nel precedente paragrafo: entrambi i protocolli seguono abbastanza bene il valore ottenuto mediante la modalità reale, anche se vi è un migliore risultato da parte della versione centralizzata.

Ciò che dal punto di vista delle RSU è importante ai fini di una corretta visione delle condizioni del traffico è la possibilità di osservare l'andamento oscillatorio di

tale valore per comprendere quando e con quale modalità i veicoli si stanno accodando in attesa di accedere all'incrocio oppure stanno transitando e svuotando la carreggiata.

Osservando nel dettaglio le tracce ottenute nei grafici presentati in precedenza in Figura 5.7, è possibile distinguere le due fasi principali che i veicoli attraversano nel corso del loro passaggio: in un primo momento i valori salgono lentamente ad indicare che i veicoli stanno rallentando o sono fermi in attesa del segnale verde emesso dal semaforo; successivamente le curve scendono velocemente indicando che i veicoli hanno ripreso la loro marcia e stanno liberando il tratto stradale osservato.

Durante la prima delle due fasi entrambi i protocolli seguono l'andamento del numero reale di veicoli che si stanno avvicinando in modo abbastanza preciso. Nella seconda fase la discesa della curva del protocollo decentralizzato è più ripida rispetto a quella centralizzata, che segue fedelmente la traccia reale; questo è comprensibile se si considera che nel momento in cui il leader di un gruppo raggiunge la distanza limite dalla RSU e ne comanda la terminazione vi è un grande quantitativo di nodi che vengono simultaneamente disattivati.

Nei grafici della Figura 5.11 sono mostrati altri risultati ottenuti prendendo i dati sulla velocità dei nodi per le direzioni 68° e -8° di A. Costa e quelle 70° e 153° di Pasubio. Una stima della velocità assunta dai nodi in una direzione di interesse è un fattore critico per comprendere appieno le attuali condizioni del traffico. Attraverso i due indicatori di quantità e di velocità la RSU è in grado di stimare rispettivamente la portata e la rapidità di scorrimento del flusso di veicoli in transito.

Nel corso delle due fasi precedentemente introdotte, i valori della velocità tendono inizialmente a scendere lentamente a partire da un picco di massimo locale, indicando in tal modo che i veicoli stanno arrestando la loro corsa mentre si accodano al semaforo; la curva mantiene valori prossimi allo zero fintanto che ai veicoli viene interdetto l'accesso all'incrocio e risale velocemente quando questi riprendono a scorrere.

Per accentuare questo comportamento è stata aggiunta una traccia ulteriore nei grafici (mostrata in viola) ottenuta dai valori di velocità raccolti dal protocollo centralizzato escludendo tutti i nodi che hanno una distanza superiore ai 60 metri dall'incrocio. Soprattutto nella direzione -8° di A. Costa si può notare come questo indicatore aumenti molto più velocemente rispetto al dato complessivo ottenuto dalla media dei valori ricevuti da ogni veicolo, per poi diminuire con altrettanta rapidità non appena il semaforo comanda lo stop del flusso del traffico per quella direzione. Nelle altre direzioni si ha un comportamento simile anche se meno evidente.

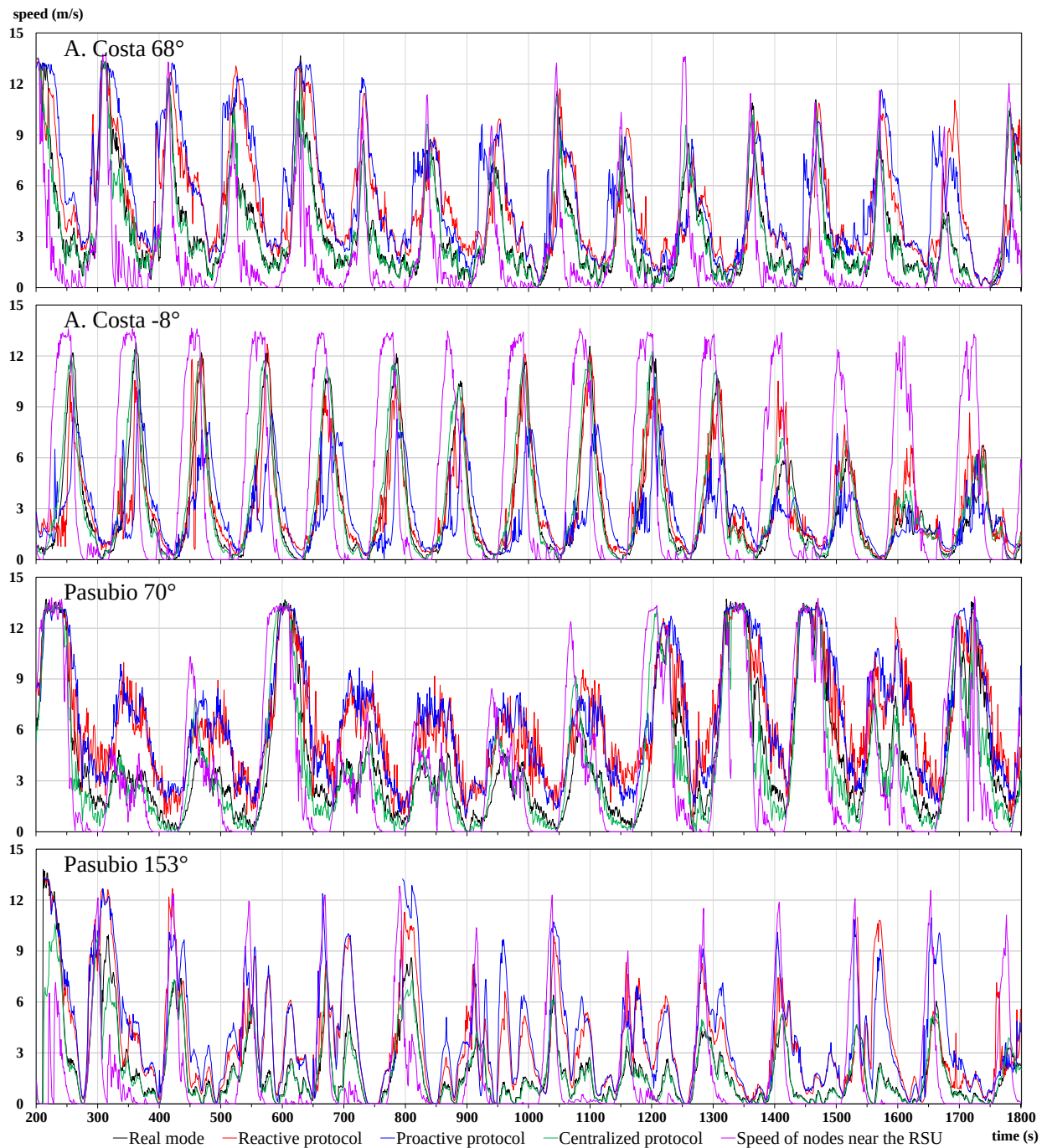


Figura 5.11: Confronto dei due protocolli per gli scenari di A. Costa e Pasubio per le direzioni 68°, -8°, 70° e 153°. I grafici indicano la velocità dei veicoli rilevata da ciascuna versione e quelli ottenuti attraverso la modalità reale. Nel protocollo decentralizzato sono presenti entrambe le modalità di formazione dei gruppi. È presentato anche un indicatore, in viola, che considera il contributo dei soli nodi vicini alla RSU fra quelli rilevati dal protocollo centralizzato.

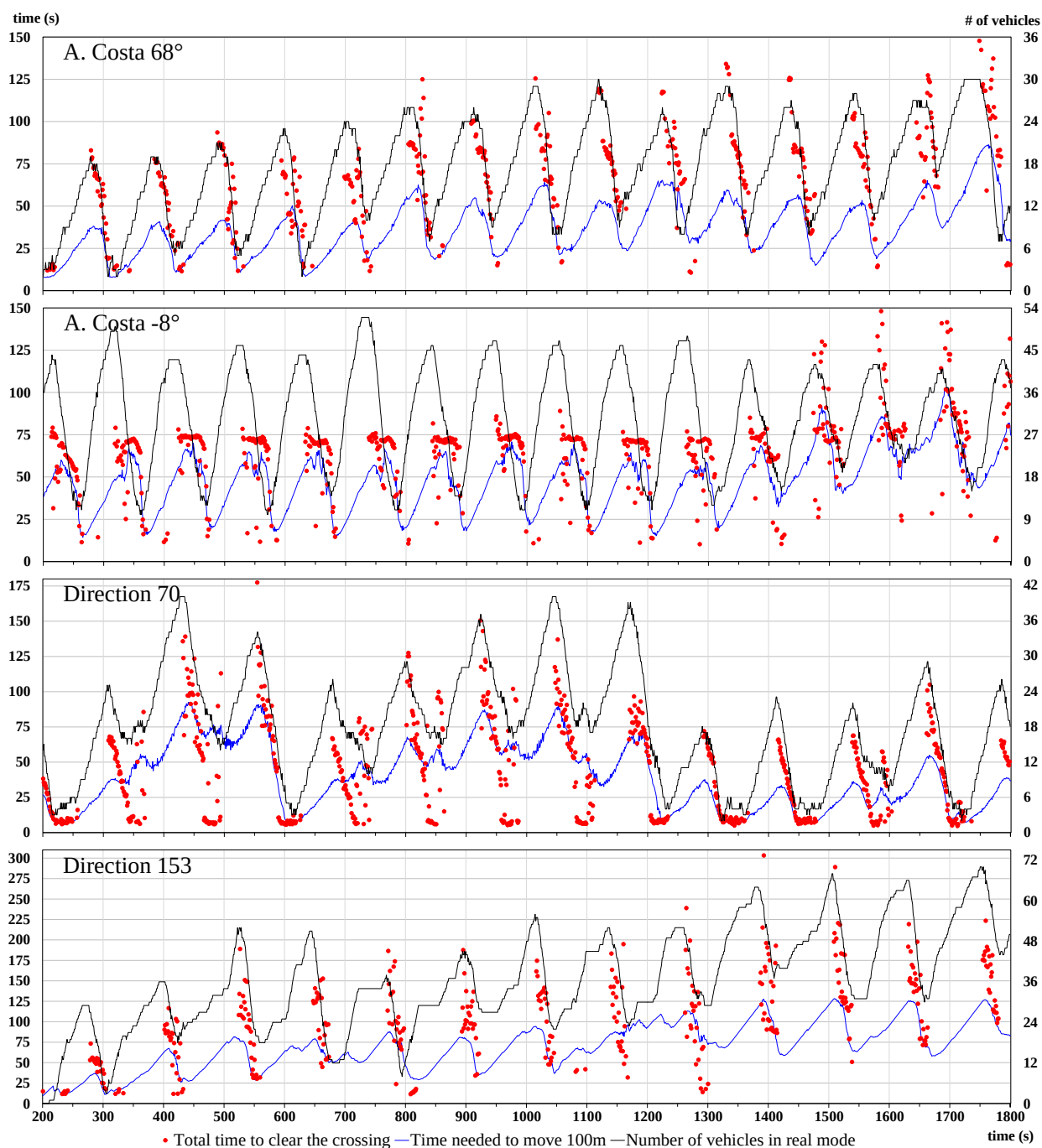


Figura 5.12: Dati sul tempo ottenuti dal protocollo centralizzato negli scenari di A. Costa e Pasubio per le direzioni 68°, -8°, 70° e 153°. I grafici indicano il tempo totale impiegato dai nodi per attraversare l'incrocio (i puntini rossi) ed il tempo medio che questi impiegano per muoversi di 100 metri (la linea blu). Come riferimento è riportato il numero di vetture presenti ottenuto nella modalità reale.

Questo indicatore filtra il contributo dei nodi più lontani e può fornire alla RSU una stima sulle condizioni del traffico nelle sue arterie di uscita, infatti nel caso questo valore sia lento a risalire a seguito di un segnale verde per la direzione su cui è calcolato, si può supporre che vi sia un intasamento successivo all'incrocio.

Un'altra informazione utile ricavata da questi dati è una stima sulla lunghezza della coda: se questo valore raggiunge lo zero la coda di veicoli supera la distanza imposta come filtro. Una logica volta a sfruttare questa informazione può creare numerosi indicatori differenti filtrando i nodi a diverse distanze dalla RSU (ad esempio quelli fra 60 e 90 metri, quelli fra 90 e 120 metri, etc.) per ottenere una stima migliore sulla lunghezza della coda che i veicoli vanno formare nel momento in cui si fermano in un incrocio in attesa di poter procedere (tailback) [Ble12]. L'estrazione di queste informazioni è possibile esclusivamente attraverso il protocollo centralizzato, in quanto il raggruppamento dei veicoli utilizzato dall'altro protocollo fa perdere le loro informazioni individuali.

La fedeltà dei risultati ottenuti rispetto alle rilevazioni effettuate in modalità reale è, per quanto riguarda il protocollo centralizzato, molto buona ed in molti tratti è praticamente indistinguibile dal riferimento. Nel protocollo decentralizzato i risultati sono generalmente accettabili, ma si assiste spesso ad alcuni ritardi ed imprecisioni nella segnalazione dei valori. In particolare, i valori riportati soffrono spesso di una sovrastima del valore riportato rispetto ai dati reali, a causa del contributo di nodi che sono a distanze maggiori rispetto al limite considerato per il calcolo dei valori di riferimento. Confrontando le due modalità di formazione dei gruppi non si notano differenze sistematiche fra le due tracce, a parte un lieve aumento della sovrastima dei valori ottenuti nella modalità proattiva. A parte questa differenza i risultati raccolti sono pressoché analoghi.

I grafici mostrati in Figura 5.12 contengono tre differenti tracce per le direzioni 68° e -8° di A. Costa e quelle 70° e 153° di Pasubio. In rosso è mostrato il tempo totale impiegato dai nodi per attraversare l'incrocio, calcolato come la differenza tra il tempo di ricezione del primo e dell'ultimo messaggio di un particolare nodo dalla RSU. L'ultimo messaggio è quello nel quale il veicolo indica il superamento della soglia di 20 metri dal centro dell'incrocio. Le informazioni così ricavate sono abbastanza scarse, in quanto la logica ottiene questi dati solamente in corrispondenza del superamento della soglia da parte di un nodo, infatti queste sono ricevute esclusivamente nei momenti nei quali è presente un segnale di verde per la direzione considerata.

Per estrarre un maggior numero di dati dal tempo di ricezione dei messaggi viene introdotto un altro indicatore, mostrato in blu nei grafici, che è ottenuto dalla RSU alla ricezione di ogni trasmissione come la differenza fra l'istante di simulazione attuale e quello nel quale è stato ricevuto per la prima volta un messaggio da quel particolare veicolo. Questa differenza viene poi normalizzata moltiplicando il valore ottenuto per 100 metri e dividendolo per la distanza tra la posizione attuale del nodo e quella che aveva all'invio del primo messaggio. Per evitare possibili errori derivanti da spostamenti troppo piccoli sono considerate valide solamente distanze superiori a 75 metri. In altre parole questo valore rappresenta il tempo medio che i nodi impiegano per muoversi di 100 metri. L'ultima traccia riportata in nero nei grafici, rappresenta il reale numero dei veicoli rilevato attraverso la modalità reale. Questi dati sono inseriti come riferimento e sono riportati sul un secondo asse a destra.

Come si può vedere questo nuovo indicatore introdotto ha un andamento oscillatorio che segue in maniera abbastanza fedele l'andamento del numero di veicoli realmente presenti nelle vicinanze della RSU. La variazione di questo dato, proporzionale alla quantità di nodi, offre alla sistema un'altra fonte per stimare questo valore che si rivelerà particolarmente importante nel caso di basse penetrazioni, come sarà esposto nel paragrafo 5.4.6.

I dati ricavati in questa figura si possono ottenere esclusivamente dal protocollo centralizzato, dato che, come nel caso dell'indicatore sulla velocità introdotto in precedenza, il raggruppamento dei veicoli fa perdere le loro informazioni individuali. Questa versione del protocollo, oltre a fornire generalmente migliori risultati rispetto a quello decentralizzato, si rivela più adattabile all'evoluzione delle euristiche di manipolazione dei dati.

5.4.5 Differenti classi di veicoli

I risultati esposti fino a questo punto prevedevano una popolazione di nodi composta esclusivamente appartenenti alla classe full. Nell'analisi seguente si vuole verificare come varia il comportamento del sistema a seguito dell'introduzione di una certa percentuale di veicoli di classe media. Per ridurre il numero di tracce presentate e viste le poche differenze riscontrate, si utilizzerà esclusivamente la modalità reattiva all'interno del protocollo decentralizzato.

Le Tabelle 4 e 5 dell'Appendice A mostrano le differenze nei valori degli attributi che differenziano le due classi di veicoli, ottenute intervenendo sul modulo di campionamento introdotto nel paragrafo 3.2.2 che consistono in una diminuzione

della precisione in termini di posizione, direzione e velocità rilevate dal nodo. Il range di trasmissione massimo è inoltre abbassato a 100 metri attraverso l'apposito parametro nella sottoscrizione per l'invio dei messaggi all'interno di iCS. Come illustrato nel paragrafo 2.4, vengono presi questi accorgimenti per simulare veicoli le cui funzionalità di comunicazione e posizionamento sono realizzate da dispositivi portatili con prestazioni inferiori rispetto a quelle dei sistemi V2X dedicati.

Le Figure 5.13 e 5.14 mostrano i risultati di cinque diverse configurazioni eseguite variando di volta in volta la composizione della popolazione di veicoli utilizzata. Ai due estremi sono rappresentati dalle linee rosse ed arancione che rappresentano rispettivamente popolazioni composte da soli veicoli full e medi. Le figure riportano i valori relativi al numero di nodi rilevati ed alle loro velocità rispettivamente nella direzione -8° di A. Costa e 153° di Pasubio. Per migliorare la leggibilità dei grafici proposti i risultati dei due diversi protocolli sono riportati separatamente e si è scelto di mostrare esclusivamente il comportamento della modalità reattiva nel protocollo decentralizzato. Il ricorso a strategie proattive ha evidenziato una serie di dati del tutto simile a quella di seguito discussa, pur con alcune occasionali irregolarità dovute alle già citate imperfezioni di tale metodo di raggruppamento. Inoltre nei grafici delle velocità mancano le tracce appartenenti ai mix di popolazione intermedi ($80\%+20\%$ e $20\%+80\%$) per aumentarne la chiarezza, il loro contributo risulterebbe comunque intermedio rispetto ai risultati riportati.

Ad una veloce analisi è possibile notare come i valori che vengono maggiormente influenzati dal passaggio da una classe all'altra risultano essere quelli relativi al numero di veicoli registrati dai gruppi: le popolazioni con prevalenza di veicoli full registrano quasi ovunque valori più elevati, mentre l'uso di nodi di classe inferiore porta a sottostimare il reale numero di veicoli nell'area interessata.

Questo risultato è del tutto prevedibile e risulta facilmente giustificabile considerando la minore portata dei veicoli di fascia media che preclude loro la possibilità di formare gruppi molto vasti e soprattutto di comunicare con la RSU a distanze superiori a circa 100 metri. Occorre inoltre ricordare che in tutti i test effettuati fino a questo momento il rilevamento dei veicoli in modalità reale è impostato per tracciare nodi ad una distanza compresa tra 20 e 170 metri, il che comporta un notevole svantaggio per i membri della classe meno performante.

Lo studio delle diverse curve relative ai valori di velocità media non evidenzia grosse differenze tra le diverse classi. La tendenza dei dispositivi meno performanti a

sovrastimare il reale valore di velocità, visibile soprattutto nei dati relativi al protocollo centralizzato, è causato ancora una volta dal limite del loro raggio di trasmissione. Durante la fase di svuotamento del tratto stradale, i valori di velocità consegnati alla RSU coprono quindi soltanto quei nodi più vicini ad essa i quali riescono a riacquistare velocità con maggiore anticipo rispetto a quelli più distanti. I risultati ottenuti sono quindi simili a quelli mostrati in Figura 5.11 nella quale era stata aggiunta una traccia per accentuare queste oscillazioni di velocità ottenuta filtrando il contributo dei nodi più lontani.

Passando ad un confronto fra i risultati ottenuti dai due protocolli si nota, anche in questo caso, una maggiore stabilità dei dati raccolti dal protocollo centralizzato, rispetto al risultato in generale più caotico ottenuto da quello decentralizzato. Le prestazioni ottenute da quest'ultimo protocollo sono in ogni caso inferiori o al massimo pari a quelle ottenute da quello centralizzato anche nei casi in cui il numero di veicoli medi sia la maggioranza, nonostante la minore dipendenza dei risultati dal raggio di trasmissione delle interfacce utilizzate.

Analizzando i grafici si può notare con grande chiarezza l'incidenza della limitazione del range trasmissivo nel protocollo centralizzato, soprattutto nel grafico relativo allo scenario di A. Costa. Il comportamento di queste curve è del tutto logico: le tracce salgono in maniera abbastanza costante quando i veicoli iniziano ad accodarsi fino a quando la lunghezza della coda non comincia a superare il range trasmissivo dei veicoli. A questo punto i risultati iniziano a discostarsi a causa del minore numero di veicoli che riescono a comunicare nelle posizioni più lontane. Questo comportamento è visibile soprattutto nel caso di totale assenza di veicoli della classe full, dato che il numero di veicoli sale fino a quando la coda non supera il raggio di trasmissione, per poi bloccarsi e rimanere costante fino a quando i veicoli non iniziano a defluire. Questo comportamento è maggiormente visibile nella direzione -8° di A. Cosa a causa della differenza nel numero di carreggiate rispetto alla direzione 153° di Pasubio (3 invece di 4): il numero di veicoli arriva a circa 30 unità prima di livellarsi, mentre la corsia aggiuntiva dell'altro scenario permette di raggiungere circa 50 unità.

I dati ottenuti dal protocollo decentralizzato mostrano un comportamento simile, anche se meno evidente a causa di rumore ed interferenze, soprattutto nello scenario Pasubio, dove i risultati non rispecchiano esattamente le variazioni nella concentrazione delle due diverse classi di veicoli.

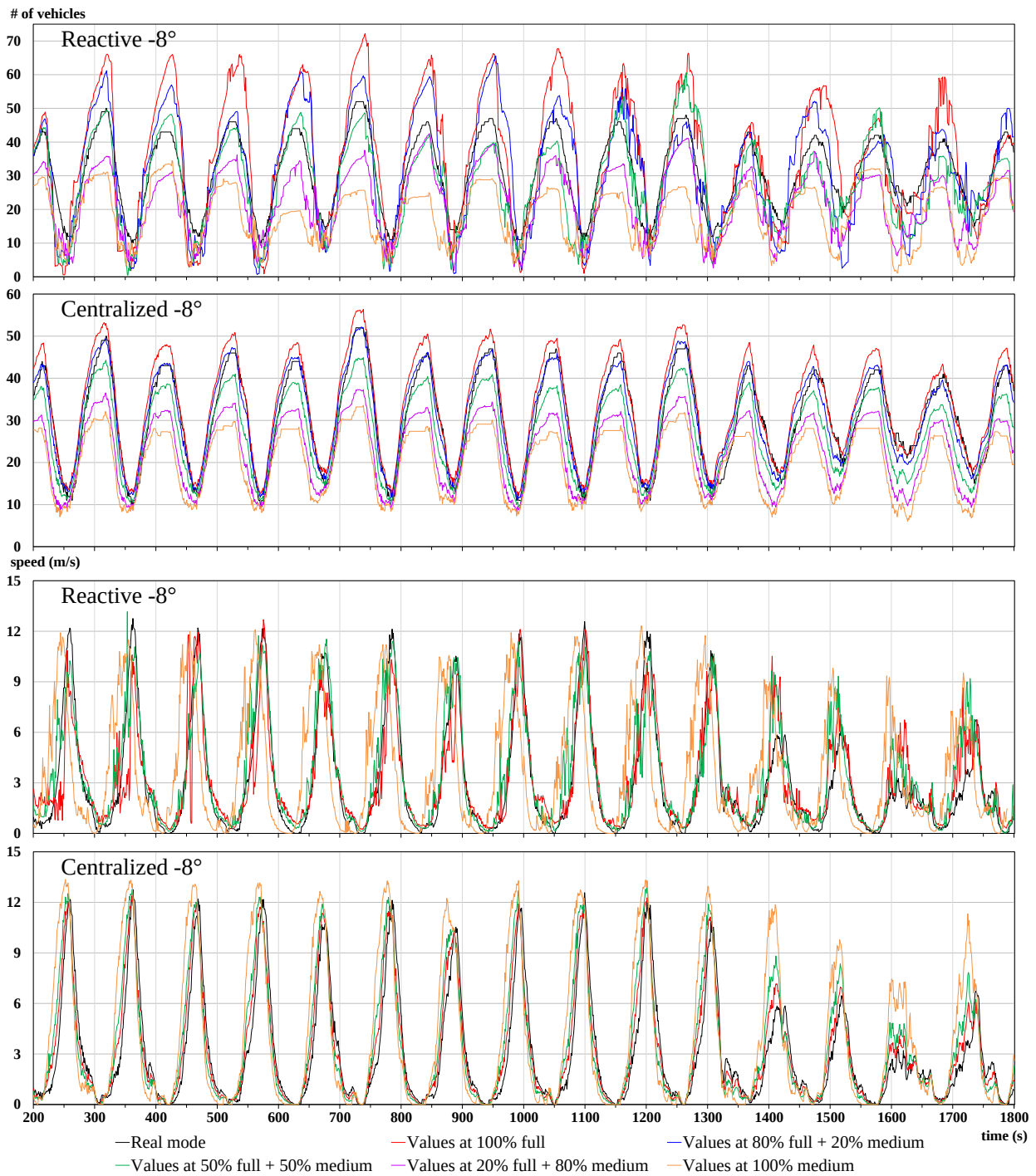


Figura 5.13: Confronto dei due protocolli per la direzione -8° dello scenario di A. Costa nel caso di differenti mix di popolazione dei veicoli. La prima coppia di grafici indica il numero di veicoli rilevati dai due protocolli ed il reale valore ottenuto dalla modalità reale. La seconda coppia indica la velocità dei veicoli rilevata da ciascuna versione e quelli ottenuti attraverso la modalità reale.

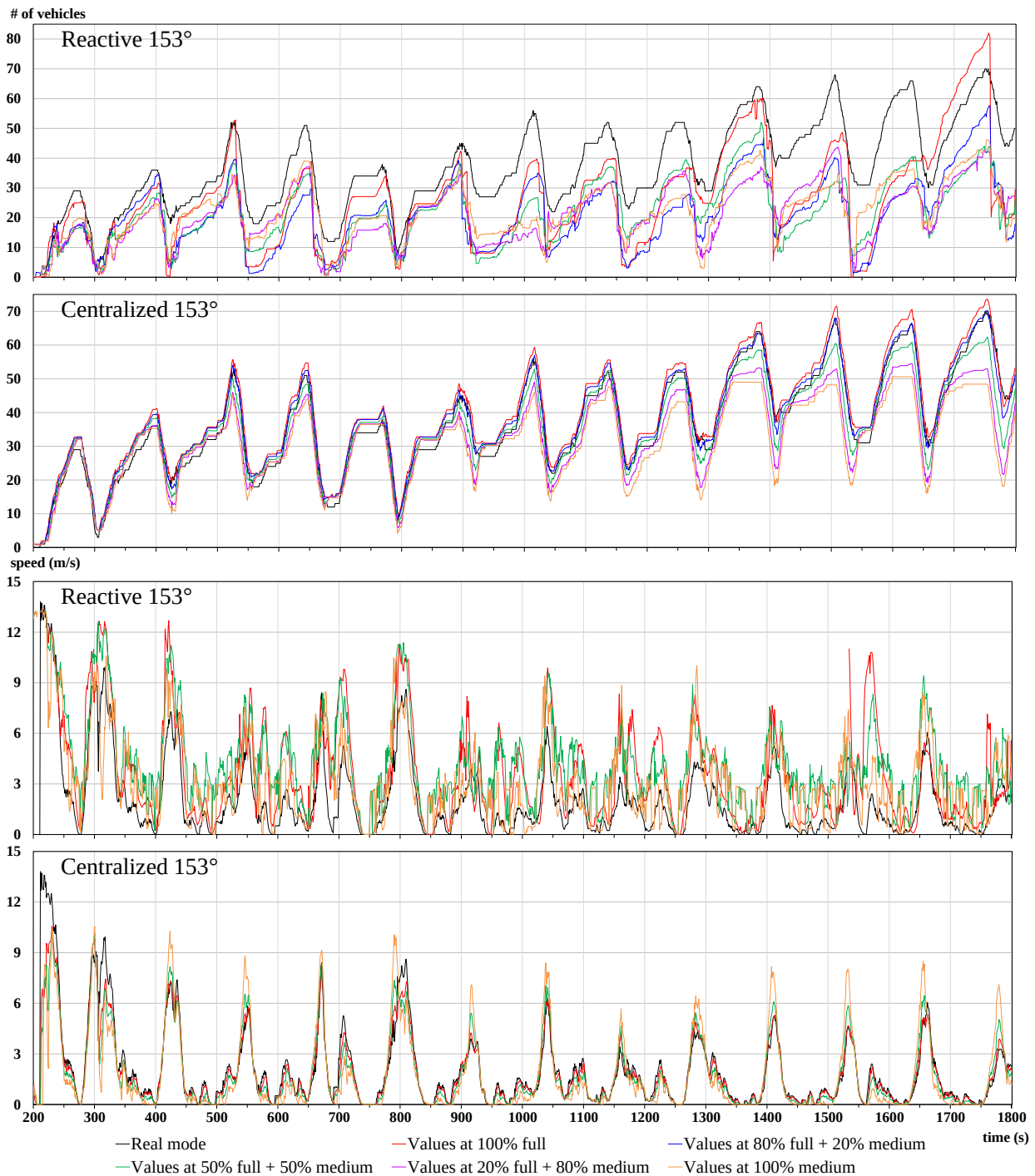


Figura 5.14: Confronto dei due protocolli per la direzione 153° dello scenario di Pasubio nel caso di differenti mix di popolazione dei veicoli. La prima coppia di grafici indica il numero di veicoli rilevati dai due protocolli ed il reale valore ottenuto dalla modalità reale. La seconda coppia indica la velocità dei veicoli rilevata da ciascuna versione e quelli ottenuti attraverso la modalità reale.

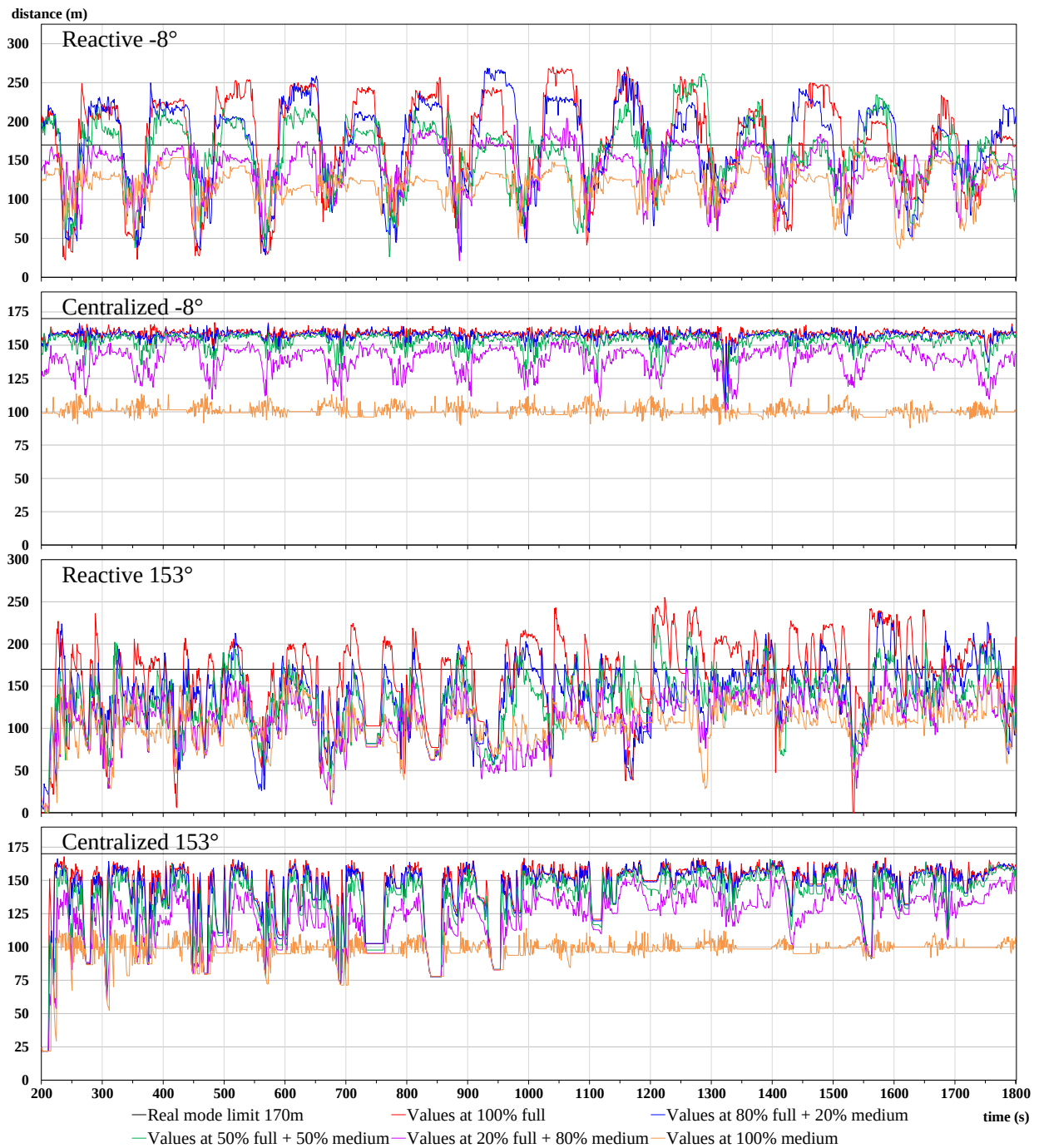


Figura 5.15: Confronto dei due protocolli negli scenari di A. Costa e Pasubio per le direzioni -8° e 153° nel caso di differenti mix di popolazione dei veicoli. I grafici mostrano la distanza del veicolo più lontano rilevata dal protocollo in ciascuna versione ed il limite della modalità reale di 170 metri.

Il confronto dei protocolli sui grafici che rappresentano i dati sulla velocità mostrano una situazione simile a quella descritta nel precedente paragrafo: il protocollo centralizzato raccoglie generalmente risultati che seguono in maniera più federe l'andamento reale, mentre quello decentralizzato introduce rumore ed alcuni ritardi, riuscendo però a fornire risultati generalmente accettabili.

La Figura 5.14 riportata le distanze del veicolo più lontano che prende parte alla raccolta di dati. Nei grafici relativi al protocollo centralizzato si nota molto chiaramente il limite di 100 metri imposto per i veicoli medi visibile soprattutto nella traccia arancione, che rappresenta tale valore nel caso di totale assenza di tipologia full. Il protocollo decentralizzato risponde meglio alle variazioni del mix di popolazione grazie alla possibilità dei veicoli di comunicare direttamente tra di loro che permette di limitare l'influenza del minore raggio trasmissivo. Nonostante questo è però chiaramente limitato dalla minore area di copertura dei segnali.

La possibilità di monitorare veicoli in un'area più estesa sembra pertanto essere l'unico reale svantaggio dovuto all'utilizzo di dispositivi di fascia media. Nel complesso, l'insieme dei protocolli sviluppati riesce comunque a portare a compimento le funzionalità critiche per il sistema ed a fornire una buona stima dello stato del traffico, pur essendo quest'ultima inevitabilmente ristretta ad un'area più ridotta, soprattutto nel caso del protocollo centralizzato.

Per contrastare le limitazioni dovute alle interfacce di comunicazione dotate di uno scarso raggio operativo è possibile in ogni caso estendere l'architettura del sistema per fare in modo che sia possibile consegnare alla RSU i dati prodotti da eventuali veicoli fuori dalla sua portata, in modo che il loro contributo possa comunque essere considerato. Per realizzare tale funzionalità è necessario nel protocollo decentralizzato predisporre una forma di comunicazione e collaborazione tra gruppi conformanti vicini al fine di trasportare le informazioni relative ad un'area di interesse verso dei nodi situati a distanze altrimenti irraggiungibili; nel protocollo centralizzato sarà invece necessario introdurre un protocollo di routing dei messaggi che garantisca la comunicazione di nodi a distanze superiori rispetto al loro raggio di comunicazione. Entrambe queste proposte saranno suggerite nel paragrafo 5.5.3.

5.4.6 Basse penetrazioni

Nel campo dei sistemi mobili, un indicatore di grande interesse nello studio di nuovi protocolli riguarda la performance del sistema in esame per diversi fattori di penetrazione. In altre parole, è spesso desiderabile che l'analisi di una tecnologia che

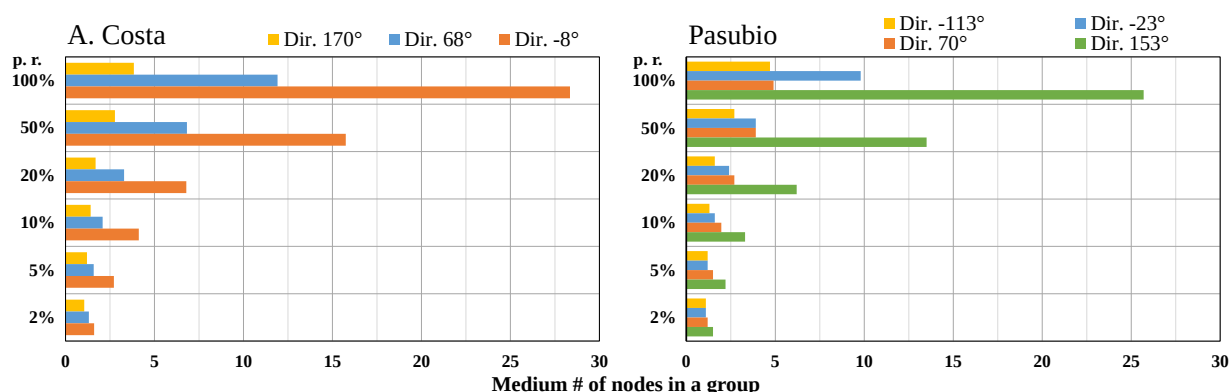


Figura 5.16: Dimensione media dei gruppi negli scenari di A. Costa e Pasubio per valori penetrazione decrescenti. I risultati sono ottenuti dalla modalità di formazione reattiva del protocollo decentralizzato.

non è ancora emersa nel mercato consumer includa una verifica di come il prodotto è in grado di comportarsi quando esso non è ancora stato adottato da una grossa quantità di utenti. Nel caso in questione si vuole comprendere quali performance è possibile aspettarsi dall'architettura presentata nei precedenti capitoli quando questa si trova ad operare in un ambiente caratterizzato da percentuali di adozione significativamente inferiori al 100%.

Per presentare questo tipo di test in maniera concisa e facilmente leggibile si è scelto di concentrare l'attenzione su un'unica direzione di per entrambi gli scenari: -8° in A. Costa e 153° in Pasubio e ci si è limitati a riportare i soli risultati derivati dall'utilizzo della modalità reattiva nel protocollo decentralizzato. Questa scelta è giustificata dal fatto che analoghi studi condotti sulle altre direzioni ed anche utilizzando la strategia proattiva hanno mostrato risultati del tutto equivalenti a quelli di seguito esposti. Le impostazioni utilizzate sono le stesse utilizzate in precedenza ma si è qui proceduto a svolgere sei classi di simulazioni ciascuna con un numero di veicoli shadow (cioè sprovvisti del sistema ITS) via via crescente. Questi nodi sono scelti casualmente in maniera uniforme all'interno della popolazione totale dei veicoli simulati. I fattori di penetrazione considerati sono pari al 100%, 50%, 20%, 10%, 5% e 2%. Sono stati utilizzati esclusivamente veicoli di classe full per rappresentare i nodi provvisti di capacità di comunicazione.

Durante la definizione del protocollo centralizzato nel paragrafo 4.2.1 è stata fornita come motivazione principale per l'abbandono del raggruppamento dei veicoli la riduzione della loro utilità in scenari caratterizzati da basse penetrazioni a causa della riduzione del numero medio di nodi presenti al loro interno. In Figura 5.16 è mostrato questo andamento per ogni direzione di entrambi gli scenari in funzione della

penetrazione. Dai valori ottenuti si può osservare la grande dipendenza di questo valore a queste variazioni: sotto al 10% il numero di nodi all'interno di un gruppo scende a qualche unità arrivando a circa un singolo nodo per il caso limite del 2%. Viceversa si può notare la grande efficienza della strategia di formazione dei plotoni che riesce ad aggregare alcune decine di unità quando la penetrazione della tecnologia e la concentrazione dei veicoli è abbastanza alta. Come si è visto nei risultati riportati in precedenza, nonostante l'efficienza della strategia di raggruppamento, i risultati ottenuti sono comunque mediamente inferiori rispetto a quelli ottenibili dal protocollo centralizzato.

Nelle figure 5.17 e 5.18 è riportato il comportamento del sistema relativamente alle informazioni sulla quantità di nodi rilevati e sulla loro velocità rispettivamente per lo scenario di A. Costa e Pasubio. Relativamente al numero dei nodi rilevati, le curve mostrate in figura presentano un declino in proporzione diretta e perfettamente lineare al decrescere della percentuale di veicoli dotati del sistema ITS. Questo risultato è del tutto normale nel momento in cui la scelta della composizione dei veicoli avviene in maniera uniforme. Si noti in ogni caso come le curve rappresentative delle percentuali più basse presentino una forma decisamente più morbida e regolare: questa sorta di perdita di definizione è indubbiamente imputabile al progressivo decrescere del numero di nodi coinvolti nel sistema, il che comporta inevitabilmente una sensibile diminuzione dei dettagli relativi allo stato reale del traffico. Non vi è una grande differenza fra i risultati ottenuti da entrambi i protocolli: la Figura 5.19 mostra una comparazione diretta dei risultati ottenuti dalle due versioni per i fattori di penetrazione di penetrazione più bassi. Si può notare un leggero vantaggio per il protocollo centralizzato nello scenario Pasubio, che è però insufficiente per creare una differenza tangibile sulle informazioni ottenibili con questo tipo di dati raccolti.

L'effetto che le variazioni del fattore di penetrazione comportano sul rilevamento della velocità media dei nodi mostra un risultato molto interessante: le diverse curve non presentano una correlazione molto stretta e lineare con la percentuale di veicoli shadow presenti nell'ambiente. Questo fenomeno è perfettamente giustificabile se si considera che una diminuzione del numero di veicoli in grado di partecipare alle comunicazioni non influisce in alcun modo sul comportamento degli stessi in termini di posizione e velocità assunte nel tempo. Anche quando il numero di nodi in grado di scambiare dati con le RSU è piuttosto basso, le informazioni relative alla loro velocità riescono comunque a riflettere adeguatamente le condizioni complessive dell'insieme di veicoli nel loro stesso senso di marcia.

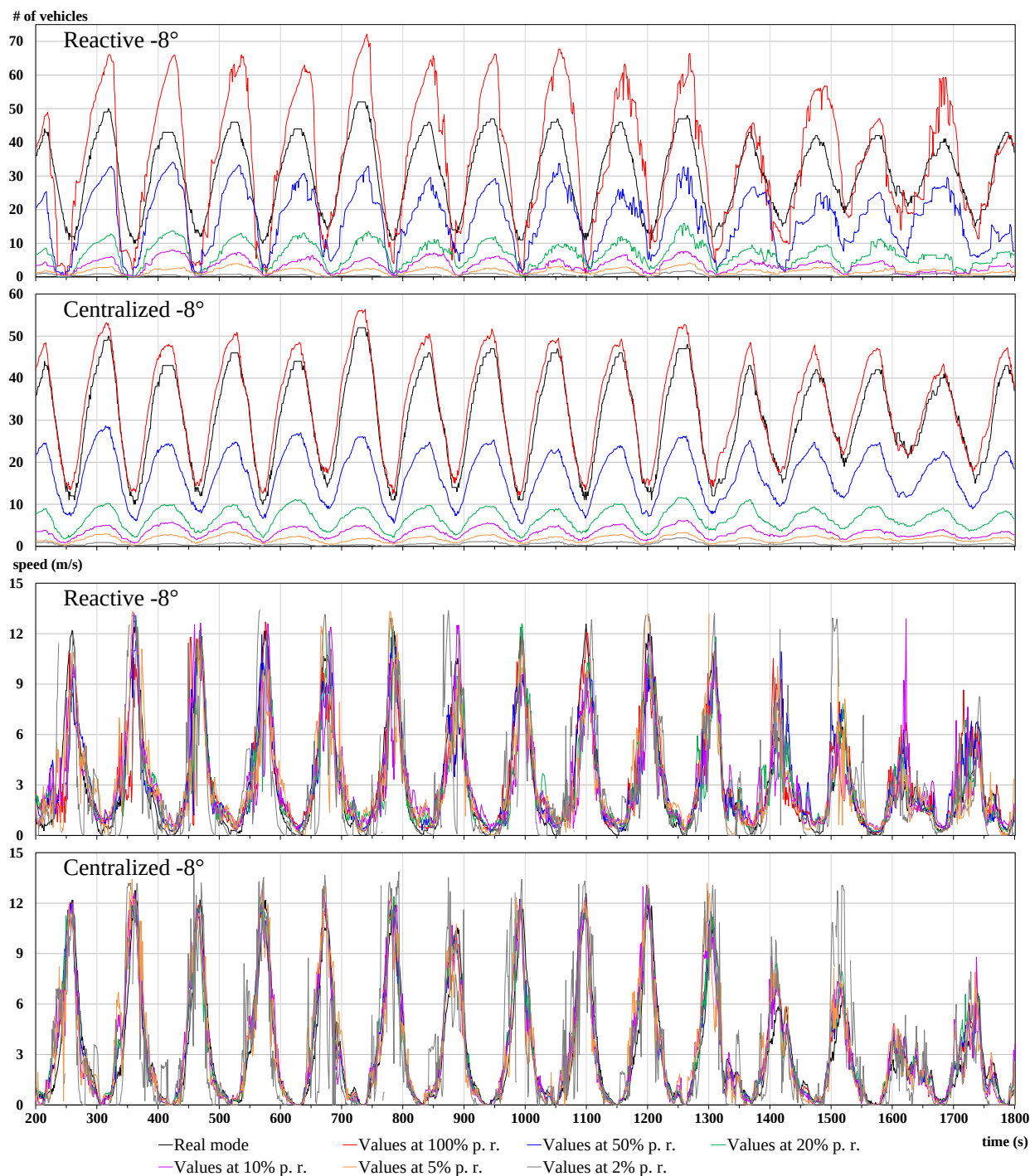


Figura 5.17: Confronto dei due protocolli per la direzione -8° dello scenario di A. Costa nel caso di differenti penetrazioni della tecnologia utilizzata. La prima coppia di grafici indica il numero di veicoli rilevati dai due protocolli ed il reale valore ottenuto dalla modalità reale. La seconda coppia indica la velocità dei veicoli rilevata da ciascuna versione e quelli ottenuti attraverso la modalità reale.

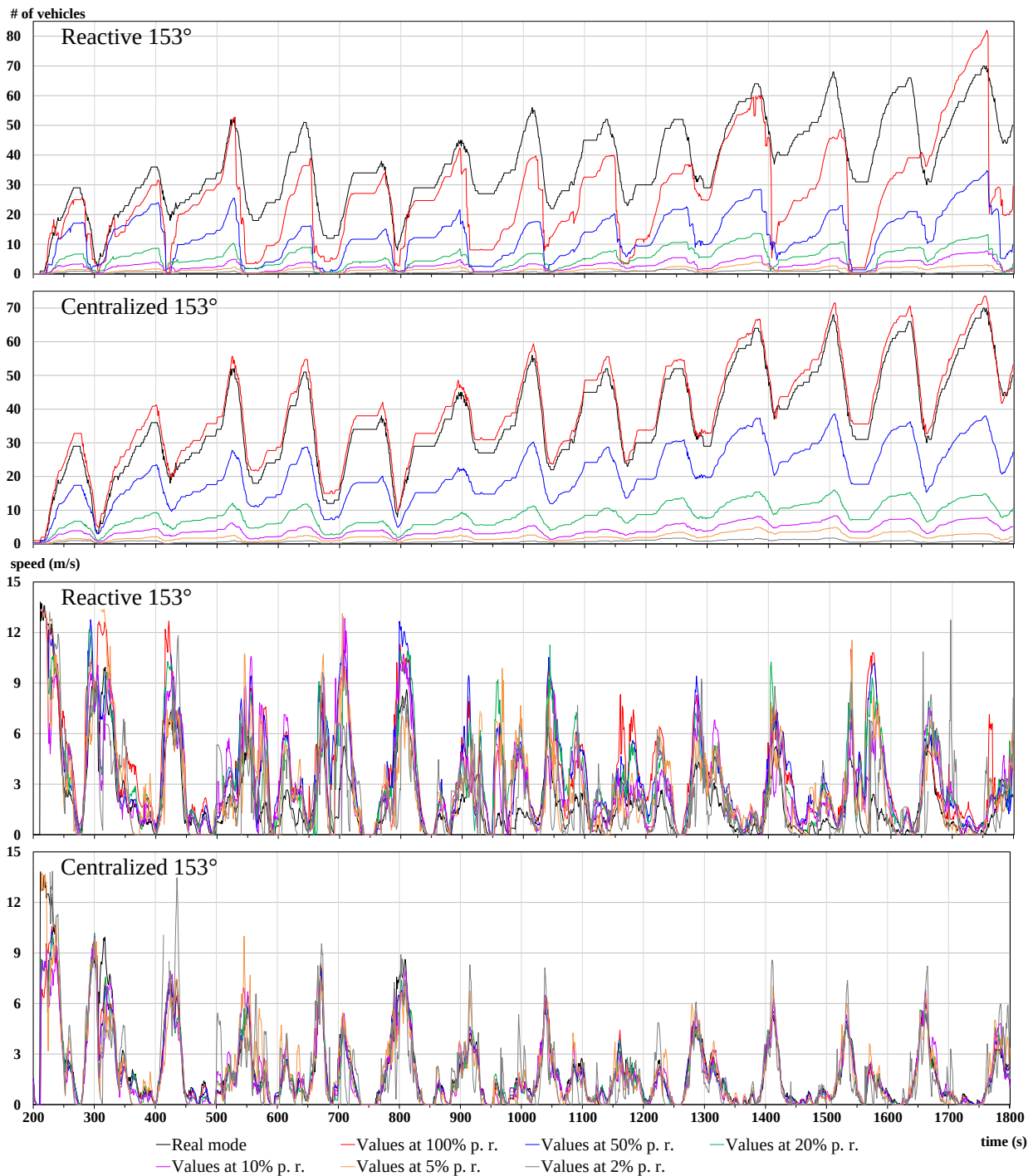


Figura 5.18: Confronto dei due protocolli per la direzione 153° dello scenario di Pasubio nel caso di differenti penetrazioni della tecnologia utilizzata. La prima coppia di grafici indica il numero di veicoli rilevati dai due protocolli ed il reale valore ottenuto dalla modalità reale. La seconda coppia indica la velocità dei veicoli rilevata da ciascuna versione e quelli ottenuti attraverso la modalità reale.

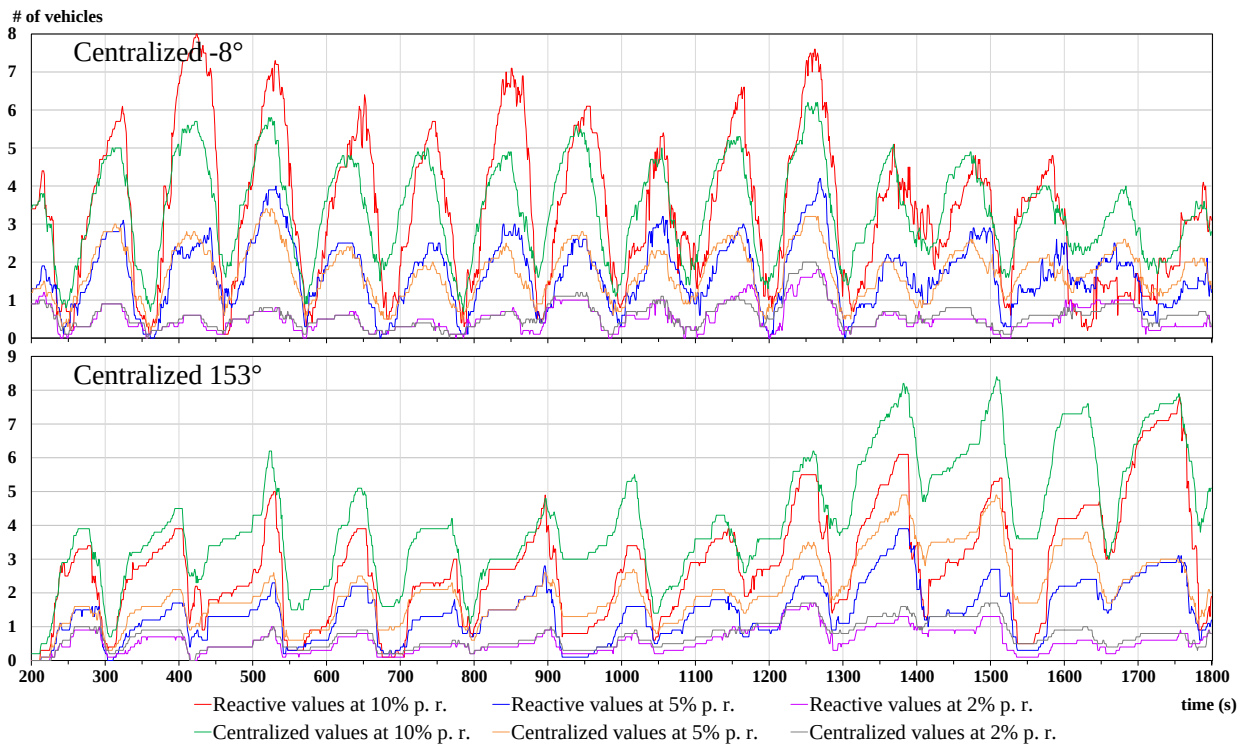


Figura 5.19: Confronto dei due protocolli negli scenari di A. Costa e Pasubio per le direzioni -8° e 153° nel caso di differenti penetrazione della tecnologia utilizzata. I grafici mostrano il numero di veicoli rilevati dai due protocolli nei casi di penetrazioni più basse.

Confrontando i due protocolli si rivela un andamento più costante dei risultati ottenuti attraverso la versione centralizzata, mentre quella decentralizzata restituisce dati più rumorosi. Entrambe le versioni hanno un andamento simile e si nota una certa tendenza a sovrastimare il valore reale nei casi di minore penetrazione.

Come è stato introdotto precedentemente, la maggior parte dei risultati riportati nei grafici di questo capitolo è ulteriormente migliorabile per mezzo di una logica addizionale installata sulle singole RSU. Anche la presenza di numerosi veicoli shadow può essere compensata a priori a patto di potere assumere come valide alcune importanti ipotesi di partenza. La prima riguarda il fatto di conoscere con buona precisione la percentuale di veicoli dotati di sistema ITS; la seconda condizione, non sempre molto realistica, richiede che la distribuzione delle due classi di nodi sia uniforme, così come è stata assunta per i test sopra presentati.

Se entrambe le queste assunzioni sono ammissibili, è possibile introdurre una semplicissima logica a livello di RSU per correggere le rilevazioni ottenute negli scenari dove la percentuale di penetrazione è conosciuta. Se per esempio è noto a priori

che la popolazione dei veicoli è composta all'80% di veicoli shadow (curve sulla penetrazione del 20% del grafico precedente) si potrà applicare un fattore correttivo pari a 5 per tentare di normalizzare la stima in proprio possesso. Ovviamente questa tecnica va applicata soltanto a quelle tipologie di dati che risentono in proporzione diretta dell'abbassamento della percentuale di adozione: i dati relativi alla velocità media ad esempio non devono essere corretti utilizzando questo principio.

La Figura 5.20 mostra gli effetti di tale operazione applicata ai dati relativi al numero di veicoli mostrati nei grafici nelle figure 5.17 e 5.18. Gli effetti della correzione sono piuttosto buoni e le varie curve tendono ad allinearsi come previsto, in particolare per percentuali superiori al 10%.

Nei casi caratterizzati da una forte congestione (particolarmente degno di nota è il picco in direzione -8° al secondo 1300) si assiste tuttavia ad una marcata sovrastima del reale numero di veicoli da parte delle curve a percentuale più bassa. La maggiore fedeltà dei dati raccolti con il protocollo centralizzato permette di ottenere risultati migliori rispetto a quelli possibili con quello decentralizzato: i risultati ottenuti sono in genere abbastanza fedeli tranne nel caso peggiore del 2% ed in alcuni punti in quello del 5%.

I grafici riportati nella Figura 5.21 mostrano il comportamento dell'indicatore sul tempo medio che occorre ai nodi per spostarsi di 100 metri, come definito nel paragrafo 5.4.4. È riportato come riferimento, mostrato in nero, il valore reale del numero di nodi ottenuto attraverso la modalità reale su un asse separato, in modo da contestualizzare le variazioni di questo indicatore. Come illustrato in precedenza l'andamento di queste tracce segue in maniera abbastanza federe la quantità di veicoli presenti e può quindi essere utilizzato per ottenere una stima di questo valore. Questi dati, ottenibili esclusivamente dal protocollo centralizzato, mostrano una grande costanza rispetto alle variazioni della percentuale di penetrazione della tecnologia utilizzata, e non presentano evidenti discostamenti fra le diverse tracce, fatta eccezione per gli errori riscontrati in alcuni momenti dai valori ottenuti nel caso peggiore caratterizzato dalla penetrazione del 2%. Questo risultato sarà confermato dai valori dell'indice delta mostrati in seguito.

La Figura 5.22 mostra l'andamento dell'indicatore delta sui dati raccolti dal protocollo centralizzato al variare della penetrazione dei nodi in gradi di prendere parte al protocollo.

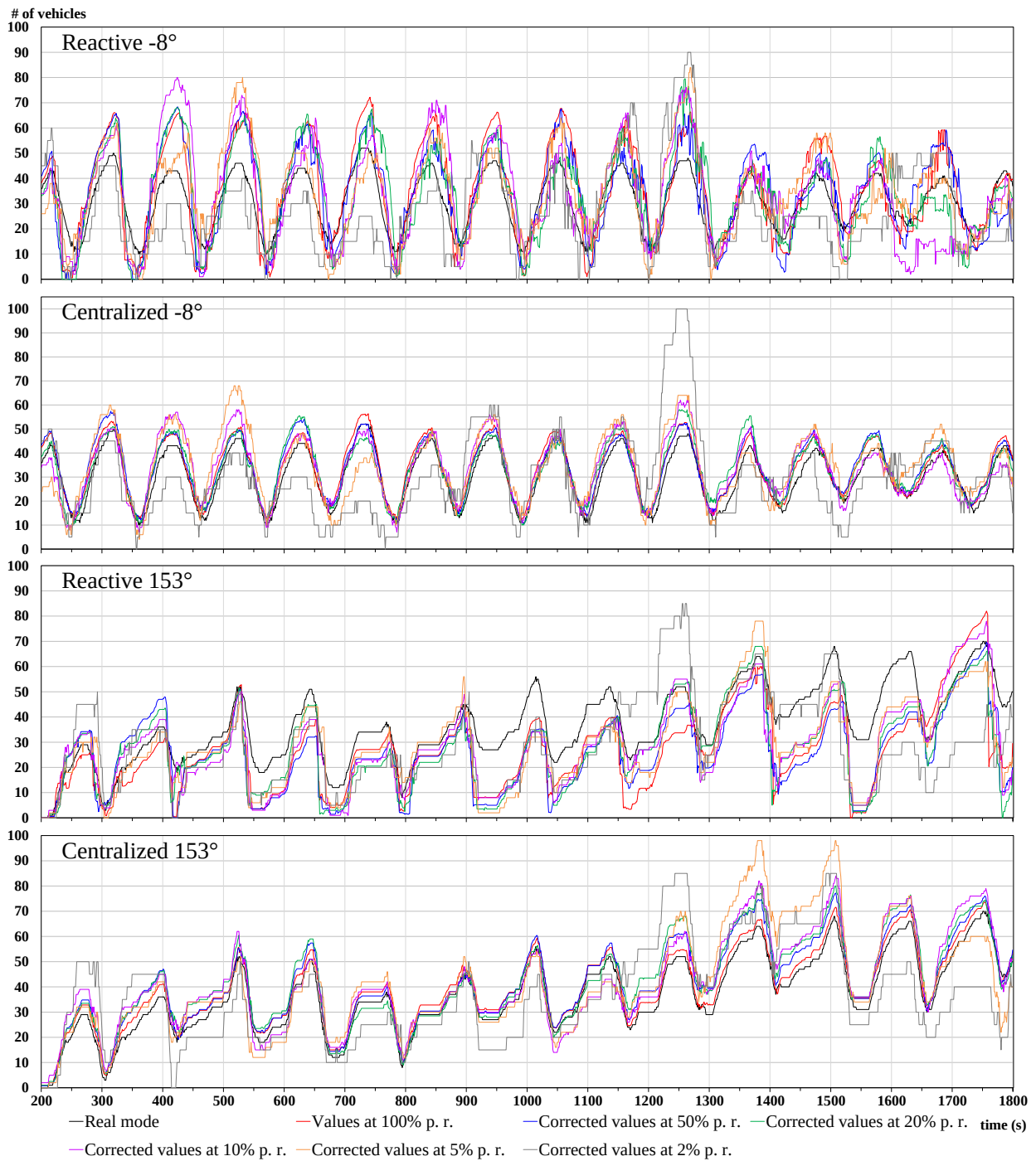


Figura 5.20: Confronto dei due protocolli negli scenari di A. Costa e Pasubio per le direzioni -8° e 153° nel caso di differenti penetrazioni della tecnologia utilizzata. I grafici mostrano il numero di veicoli rilevati dai due protocolli a seguito della correzione introdotta per compensare la minore penetrazione.

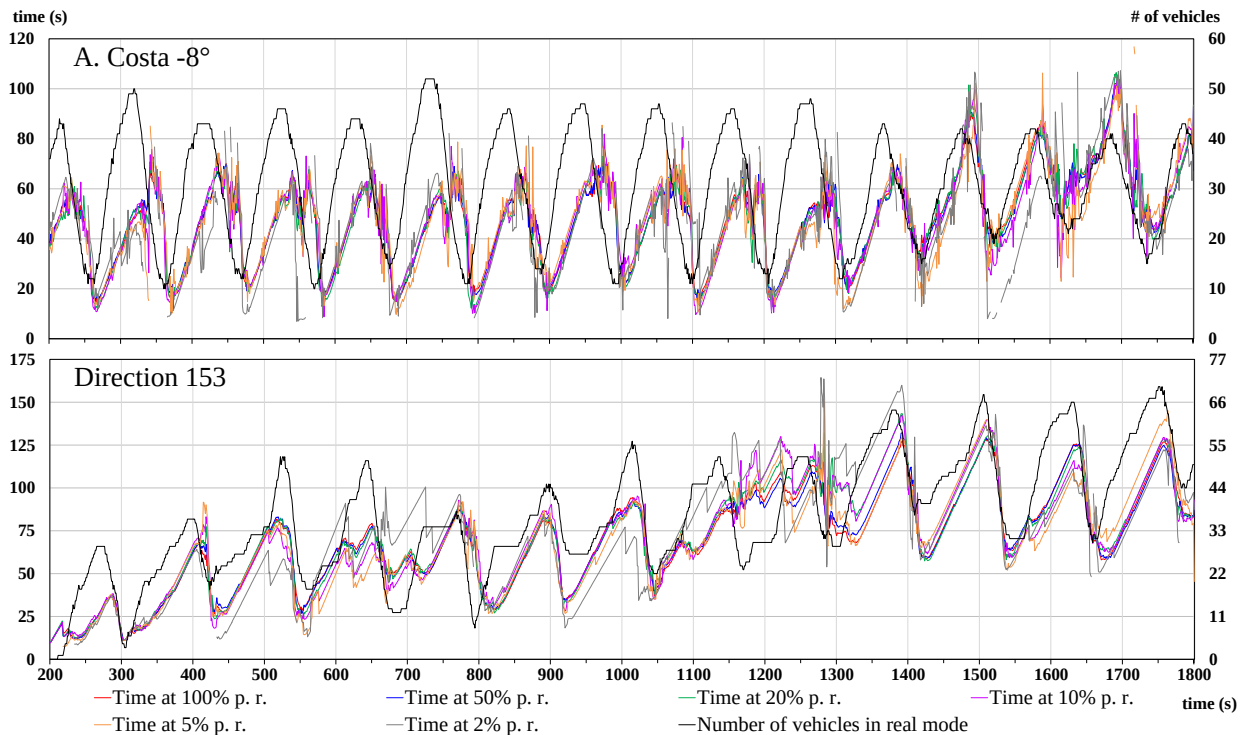


Figura 5.21: Dati sul tempo ottenuti dal protocollo centralizzato negli scenari di A. Costa e Pasubio per le direzioni -8° e 153° nel caso di differenti penetrazioni della tecnologia utilizzata. I grafici indicano medio che impiegano i nodi per muoversi di 100 metri. Come riferimento è riportato il numero di vetture presenti ottenuto nella modalità reale.

Questo valore è calcolato sulle grandezze prese in considerazione nei grafici precedenti utilizzando come riferimento i valori ottenuti nel caso di penetrazione completa. I primi due grafici si riferiscono ai valori nelle direzioni 68° e -8° di A. Costa, gli altri due a quelli nelle direzioni 70° e 153° di Pasubio. In ogni grafico sono mostrati i valori del delta calcolato sul numero di veicoli rilevati dal protocollo, su questa quantità corretta con il metodo illustrato in precedenza, sui dati riguardanti la velocità ed infine sull'indicatore del tempo medio introdotto nel paragrafo 5.4.4.

In assenza di interventi i dati relativi ai valori delta sulla quantità di veicoli mostrano un andamento decrescente lineare, che riflette in modo praticamente speculare la percentuale di nodi in grado di comunicare, come illustrato anche nelle considerazioni precedenti. A seguito delle correzioni, i valori mostrano un comportamento abbastanza costante fino ai dati che riguardano una penetrazione del 5%, sotto questa soglia inizia ad esserci una variazione tangibile rispetto ai valori della configurazione di riferimento. Le due direzioni maggiormente trafficate hanno un comportamento

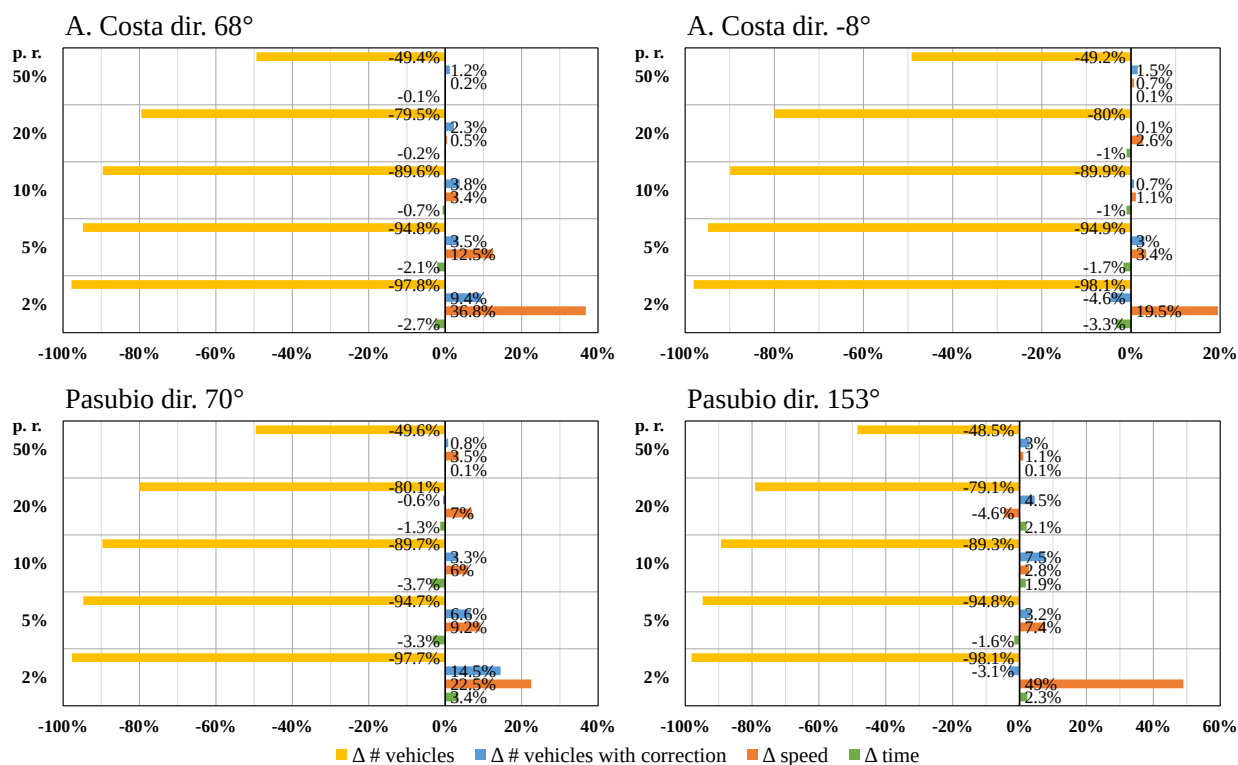


Figura 5.22: Valore delta calcolato sui dati ottenuti dal protocollo centralizzato rispetto al caso di penetrazione completa. I grafici mostrano i valori delta calcolati su varie grandezze: in giallo sul numero di vetture rilevate, in blu questo valore corretto, in arancione sulla velocità dei nodi ed in verde sul tempo medio che impiegano per spostarsi di 100 metri.

differente rispetto alle altre, infatti il maggior numero di veicoli presenti permette di ottenere una stima migliore anche nei casi di penetrazione limite.

I dati che riguardano il valore della velocità dei veicoli hanno un comportamento abbastanza regolare ed indipendente alle variazioni della penetrazione utilizzato, anche se tende ad essere presente una generale sovrastima per le penetrazioni più basse, che nei casi peggiori supera il 40%, come ad esempio nella direzione 153° dello scenario Pasubio. I dati ottenuti sono comunque molto buoni e, tranne nel caso limite considerato, il loro discostamento dai valori presi come riferimento risulta sempre inferiore al 10%.

L'ultimo valore riportato nei grafici si riferisce al valore del tempo medio; questi dati si rivelano completamente indipendenti alle variazioni della penetrazione: in ogni caso il discostamento massimo rispetto alla configurazione di riferimento è inferiore al 5%, con risultati analoghi anche per le direzioni non mostrate in questi grafici. Una logica di raccolta dati che utilizzi il valore di questo indicatore riuscirebbe ad avere

una approssimazione del traffico presente indipendentemente dal valore di penetrazione presente nell'ambiente reale.

Benché i risultati ottenuti dall'introduzione di una correzione sul numero di veicoli rilevati dal protocollo si siano rivelati molto buoni, una logica di questo tipo è di difficile applicazione in quanto le due assunzioni su cui si basa questo miglioramento sono difficilmente ottenibili nella realtà. La percentuale di veicoli non è generalmente costante e può variare sia durante il giorno che nel tempo; è inoltre quasi impossibile ottenere nella realtà una distribuzione uniforme di veicoli dotati di questa tecnologia, come invece utilizzato durante lo svolgimento di questi test. Occorre pertanto basare la raccolta delle informazioni su valori meno sensibili alla diffusione della tecnologia, come quelli ottenuti dalla velocità dei veicoli e sull'indicatore del tempo medio calcolato nel protocollo centralizzato.

Nel prossimo capitolo ci si preoccuperà di riassumere i principali difetti del sistema riscontrati durante l'ampio lavoro di test al quale esso è stato sottoposto. In relazione alle problematiche riscontrate ed a seguito dell'esperienza maturata nel corso della progettazione ed implementazione dell'architettura verranno proposte alcune modifiche e funzionalità aggiuntive per il progetto in grado di incrementarne le prestazioni e la robustezza.

Capitolo 6

Analisi dei risultati e sviluppi futuri

La motivazione principale che ha spinto alla realizzazione dello studio del sistema presentato nel corso dei capitoli precedenti è stata l'esigenza di sperimentare ed analizzare dei protocolli semplici ma comunque efficaci per la raccolta di dati dai veicoli in transito verso punti di interesse, finalizzata a trasferire verso i dispositivi infrastrutturali fissi un flusso di informazioni in grado di fornire una stima sulla composizione e sull'entità del traffico suddivisa in base alla particolare direzione percorsa dalle vetture in arrivo. La base di conoscenza prodotta dal sistema deve essere tale da permetterne il successivo utilizzo come input per l'attuazione di logiche di gestione del traffico dinamiche ed intelligenti.

I numerosi test condotti sull'architettura nel corso del suo sviluppo e in particolare quelli esposti nel capitolo precedente hanno dimostrato come il sistema sia in grado di rispondere a tali requisiti: i protocolli realizzati permettono la creazione di un flusso di dati verso le unità infrastrutturali a cui i nodi fanno riferimento anche a fronte di densità relativamente elevate, hardware eterogeneo e pattern di mobilità differenti e spesso imprevedibili. I dati così raccolti contengono informazioni sulla composizione, sulla quantità di veicoli che si stanno avvicinando e sulla loro velocità. Confrontando i valori ricavati dalla modalità reale con i risultati forniti dai protocolli alle varie RSU si è visto come questi ultimi siano in grado di fornire una stima sufficientemente accurata delle attuali condizioni del traffico circostante. I numerosi parametri che regolano le funzionalità dei diversi componenti dell'architettura discussa permettono in molti casi di controllare il comportamento delle varie logiche di protocollo, personalizzandole in questo modo alle proprie esigenze; il sistema si presta inoltre ad ulteriori possibilità di estensione e revisione, alcune delle quali verranno discusse al termine del presente capitolo.

Il lavoro di implementazione del sistema all'interno del simulatore di comunicazioni veicolari iTETRIS può essere inoltre utilizzato come base per la creazione di euristiche più complesse che sfruttino la base di conoscenza prodotta per attuare una politica di controllo del traffico, usufruendo interamente delle funzionalità messe a disposizione da questa piattaforma.

6.1 Risultati ottenuti

Oltre ad averne verificato il corretto funzionamento in base ai requisiti richiesti, i diversi test condotti nel capitolo 5 hanno consentito di comprendere in quali casi d'uso è possibile ottenere prestazioni migliori dal sistema e che tipologie di vantaggi è lecito attendersi da ciascuna delle due modalità di funzionamento implementate.

Nel paragrafo 5.4.2 si sono studiate le prestazioni del protocollo decentralizzato nella sua implementazione originale in NS-3 ed in quella realizzata in iTETRIS, ottenendo prestazioni leggermente inferiori per quest'ultima (figure 5.3 e 5.4). Si è comunque ritenuto accettabile una prestazione inferiore dovuta alla grande differenza nell'architettura dei due simulati, considerando anche il grande lavoro di ottimizzazione svolto per aumentarne le prestazioni (Figura 5.2).

Nel paragrafo 5.4.3 si è osservato che il protocollo centralizzato è generalmente in grado di ottenere prestazioni superiori rispetto a quello decentralizzato (Figura 5.7) e si è inoltre riscontrato un allineamento delle prestazioni fra le differenti strategie di formazione dei gruppi utilizzate in quest'ultimo protocollo. Analizzando il costo dei due sistemi in termini di messaggi scambiati si è rilevata una differenza fra la versione centralizzata che coordina in modo migliore le trasmissioni rispetto a quanto avviene in quella decentralizzata, che, soprattutto nella strategia proattiva, genera un gran numero di collisioni andando a degradare la qualità delle comunicazioni (Figura 5.9).

La distanza dall'incrocio a partire dalla quale il sistema riesce ad operare dipende principalmente dalla particolare conformazione del tratto stradale monitorato e dal raggio di portata dei veicoli (Figura 5.8). Nel caso del protocollo centralizzato i nodi non possono ovviamente comunicare con le RSU oltre tale soglia, mentre l'utilizzo di gruppi di veicoli in quello decentralizzato permette comunque ai leader di informare l'infrastruttura della presenza e dello stato di nodi posizionati oltre ad essa.

Nel paragrafo 5.3.4 sono stati commentati i risultati ottenuti dalle informazioni raccolte dal protocollo, ed esplorate le possibilità aggiuntive offerte dal protocollo centralizzato per estrarre un indicatore sulla velocità dei soli nodi posizionati entro una distanza massima dall'incrocio e per ricavare dati aggiuntivi considerando gli istanti di ricezione dei messaggi (figure 5.11 e 5.12).

Gli esperimenti condotti nel paragrafo 5.3.5 dimostrano che il sistema è in grado di funzionare discretamente anche ricorrendo a sensori ed interfacce appartenenti ad una classe meno performante (Figure 5.13 e 5.14). L'elemento che più di tutti com-

promette l'output dei protocolli risulta essere la portata dei singoli nodi, la quale incide ovviamente sulla capacità del sistema di formare gruppi di dimensione maggiore nel protocollo decentralizzato e sulla capacità di raccogliere informazioni relative a nodi maggiormente distanti in quello centralizzato. Quest'ultimo protocollo è maggiormente influenzato dalle limitazioni di questa classe a causa della totale assenza di politiche di routing dei messaggi oltre al raggio di trasmissione dei nodi (Figura 5.14).

Infine nel paragrafo 5.3.6 viene condotta una serie di test per verificare il comportamento dell'architettura in scenari caratterizzati da percentuali di penetrazione più bassa e per determinare con quali modalità la qualità delle rilevazioni diminuisce in relazione a tale parametro. La Figura 5.22 riassume in estrema sintesi questi risultati: in assenza di interventi, il sistema vede un deterioramento lineare del numero di nodi rilevati con conseguente grande peggioramento delle performance. Una semplice correzione proposta su questi dati permette di ottenere risultati utilizzabili anche per percentuali molto basse, a patto di ammettere delle precondizioni difficilmente realizzabili nella realtà. Considerare invece la velocità dei nodi ed il tempo da loro impiegato per attraversare l'incrocio permette di ottenere risultati migliori in quanto scarsamente influenzati dalla diminuzione della penetrazione della tecnologia e permettono il funzionamento del sistema anche per penetrazioni inferiori al 5% (Figura 5.20).

Dai risultati ottenuti in questi scenari caratterizzati da penetrazioni variabili si è rilevata la grande volatilità dei risultati che si basano sulla quantità di veicoli rilevati dai protocolli, in quanto estremamente dipendenti da questo parametro. Una logica dedicata alla gestione dinamica ed intelligente del traffico deve quindi fare preferibilmente uso di valori che hanno una minore dipendenza da queste variazioni, come la velocità dei veicoli. Il protocollo centralizzato permette inoltre elaborazioni più complesse sulle informazioni raccolte per ottenere, ad esempio, dati sul tempo necessario ai nodi per attraversare l'incrocio, che si sono rivelati come le grandezze meno influenzate dal valore della penetrazione.

6.2 Limiti del sistema

Nonostante la buona resa del sistema realizzato, esso non ha di certo la pretesa di risultare esente da difetti o punti di debolezza. Scopo del presente paragrafo è quello di discutere le limitazioni attuali dell'architettura proposta in questo testo, individuando particolari scenari e situazioni limite che possono precluderne l'utilizzo o

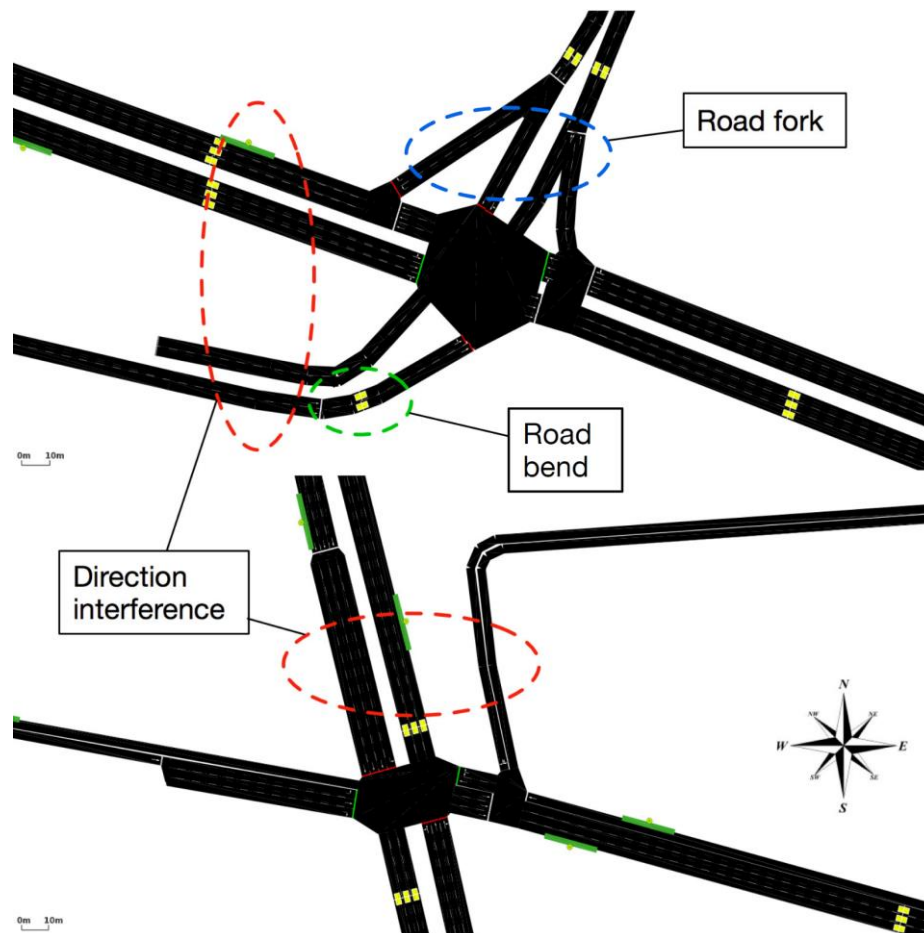


Figura 6.1: Esempi di scenari che possono presentare difficoltà di utilizzo per il sistema realizzato: i cerchi rossi indicano interferenze di direzioni tra strade contigue approssimativamente parallele, in quello blu vediamo la presenza di svincoli o deviazioni di forma atipica, il punto in verde indica una strada che effettua un brusco cambiamento di direzione poco prima dell'incrocio.

limitarne drasticamente le performance. Questa analisi verrà infine seguita da un insieme di proposte e suggerimenti su come migliorare le funzionalità del sistema attraverso ulteriori modifiche e sviluppi. Alcuni di questi miglioramenti sono stati effettivamente considerati durante la fase di progettazione ma successivamente scartati per dare maggiore spazio ad elementi più prioritari o per limitazioni di tempo.

Le maggiori difficoltà che possono presentarsi nel momento in cui si tenta di applicare l'uso del sistema a degli scenari d'uso reali sono provocate dal concetto di direzione formalizzato nel paragrafo 2.1: tale principio è infatti alla base del processo di raggruppamento dei veicoli nel protocollo decentralizzato, dell'attivazione dei nodi in quello centralizzato e permette oltretutto di differenziare i flussi di dati verso le

RSU, consentendo la loro separazione in base allo specifico percorso di provenienza. Questa strategia funziona molto bene negli incroci stradali di conformazione più classica, ovvero caratterizzati da vie con direzioni chiaramente distinte che si incontrano nel punto in cui viene posizionata l'unità infrastrutturale.

Possono esistere tuttavia numerosi casi in cui una conformazione particolarmente atipica della rete stradale comporta una serie difficoltà nel momento in cui devono essere individuati gli esatti valori di direzione che una RSU vuole monitorare. Nel seguito si farà riferimento alla Figura 6.1 per meglio comprendere i casi critici enunciati.

Interferenza tra direzioni parallele

In entrambi gli scenari presi come esempio vengono evidenziate nelle ellissi tratteggiate in rosso alcune coppie di strade appartenenti a percorsi differenti ed i cui relativi flussi di veicoli andrebbero mantenuti separati. Tali carreggiate presentano tuttavia valori di direzione quasi identici tali per cui il sistema non riuscirebbe a distinguere quali veicoli stiano effettivamente percorrendo una o l'altra strada.

Più in generale, nel momento in cui viene individuata una qualsiasi direzione di interesse per una RSU, qualsiasi percorso stradale, anche secondario oppure del tutto irrilevante per il controllo del traffico in quel particolare incrocio, caratterizzato da un valore di direzione simile o identico ad esso comporta un problema di interferenza tra due differenti strade. A seguito di tale disturbo, il sistema perde la capacità di distinguere i veicoli che transitano lungo uno qualsiasi dei due percorsi, provocando così una unione dei dati raccolti nelle due diverse strade ed alterando considerevolmente le informazioni di traffico consegnate all'infrastruttura.

Diramazioni in prossimità di incroci

Nel primo dei due esempi viene indicata in blu una particolare conformazione stradale in grado di comportare alcune difficoltà di utilizzo per il sistema realizzato: i veicoli che raggiungono l'incrocio seguendo una direzione sud-ovest si trovano a poter scegliere tra due diversi percorsi ad una distanza di poche decine di metri dal punto in cui verrebbe idealmente posizionata una RSU. Le vetture che scelgono di immettersi direttamente nell'arteria stradale principale orientata verso nord-ovest seguono per alcuni metri una direzione differente rispetto agli altri veicoli che procedono verso sud-ovest ed entrano effettivamente nell'incrocio.

A fronte di questa situazione possono presentarsi dei risultati differenti. Se la RSU non è interessata a tenere traccia dei veicoli che non entrano nell'incrocio si

troverebbe comunque in ingresso dei dati che includono il contributo dei veicoli indesiderati, almeno fino a quando questi non effettuano la svolta al punto indicato e percorrono alcuni metri. Se invece si è intenzionati a monitorare anche il tratto stradale successivo alla diramazione, nel caso di utilizzo del protocollo decentralizzato, i gruppi che andrebbero a formarsi avrebbero un ciclo di vita estremamente breve considerando che il segmento che congiunge le due vie principali ha una lunghezza di poche decine di metri; di conseguenza il flusso di informazioni che essi consegnano alla RSU di riferimento potrebbe essere incompleto o comunque estremamente frammentato a causa del breve periodo di attività concesso ai plotoni. In questo particolare caso il protocollo centralizzato non soffrirebbe di questo particolare problema, dato che non ha la necessità di formare e gestire gruppi di veicoli. Infine, se i veicoli che percorrono la diramazione non raggiungessero la distanza minima prefissata dalla RSU non concluderebbero correttamente le loro comunicazioni e questo potrebbe portare a un'ulteriore diminuzione della qualità dell'euristica.

Direzioni irregolari

Nel cerchio verde in Figura 6.1 viene indicato un punto in cui una strada di interesse presenta una forte irregolarità alcuni metri prima di confluire nell'incrocio. Tralasciando i problemi derivanti dalle interferenze tra vie parallele, i veicoli che percorrono la carreggiata subiscono un brusco cambio di direzione quando si trovano ad essere piuttosto vicini alla RSU.

Il sistema funziona generalmente bene in presenza di rettilinei sufficientemente lunghi posti immediatamente prima all'incrocio che si intende gestire; a seguito dell'irregolarità di alcune strade i veicoli potrebbero assumere direzioni non previste dalla RSU ed inattivarsi per diversi secondi oppure potrebbero iniziare a comunicare molto in ritardo, riducendo nettamente la quantità di informazioni scambiate e deteriorando di conseguenza le performance del sistema.

Anche se è possibile ricorrere ad alcune correzioni per mitigare i cali di prestazioni imputabili a percorsi stradali irregolari, il problema resta comunque di difficile soluzione e richiede certamente uno studio più approfondito.

Inefficienza del protocollo decentralizzato

I risultati presentati nel paragrafo 5.4 mostrano come il protocollo centralizzato offra generalmente risultati migliori rispetto a quelli ottenuti da quello decentralizzato, nonostante questo possa comunicare solamente nel range di copertura dei nodi. Nonostante la rimozione dei gruppi sia stata effettuata a causa della loro scarsa efficacia nelle condizioni di basse penetrazioni, questa modifica si è rilevata molto performante

anche nel caso di grande densità. Unita alla particolare attenzione posta in fase di progettazione volta alla riduzione del numero di messaggi scambiati ed al loro coordinamento ha portato ad una notevole riduzione delle collisioni rispetto a quelle ottenute nel protocollo decentralizzato.

Si è infatti riscontrata una inefficienza intrinseca dovuta alla distribuzione del controllo sullo scambio dei messaggi nel protocollo decentralizzato: non è previsto nessun tipo di coordinamento fra i vari gruppi per evitare situazioni nelle quali più leader comandino contemporaneamente una esplorazione, portando ad un picco di scambio di messaggi, da cui consegue inevitabilmente un elevato numero di collisioni fra le varie trasmissioni. Questo problema è molto alleviato nel caso del protocollo centralizzato, nel quale si è prestata particolare attenzione ad accentrare tutti i compiti di gestione in un unico punto, che può così coordinare tutte le comunicazioni in modo da evitare il più possibile la loro sovrapposizione.

6.3 Sviluppi futuri

Sono di seguito suggerite alcune possibili estensioni al sistema che se implementate potrebbero contribuire in maniera significativa ad aumentarne le performance, a migliorare la qualità dei dati ottenuti e soprattutto a rendere l'architettura più flessibile e robusta. Le strategie seguenti sono state ideate nel corso delle ultime fasi di implementazione del progetto e le limitazioni di tempo e di risorse non ne hanno purtroppo reso possibile l'inclusione nella soluzione fin qui presentata.

Ottimizzazione dei parametri di sistema

Nel corso dei capitoli 3 e 4 sono stati presentati numerosi moduli, in particolare quelli relativi alle diverse logiche di protocollo, che contengono svariati attributi di configurazione, elencati in Appendice A, mediante i quali è possibile controllare in maniera molto flessibile i parametri di funzionamento del sistema ed il comportamento dei componenti sviluppati.

Nella raccolta dei risultati esposti nel capitolo precedente sono stati utilizzati i valori di default indicati nelle rispettive tabelle, salvo ove diversamente specificato. Tali impostazioni predefinite non sono tuttavia frutto di un'estesa ricerca finalizzata alla ricerca della migliore configurazione possibile per il sistema e sono pertanto da intendersi come sperimentali. In molti casi, un attento studio sulla messa a punto dei valori di alcuni attributi chiave potrebbe rivelarsi molto proficua per ottenere prestazioni complessive più soddisfacenti.

Soprattutto per il caso del protocollo decentralizzato, uno studio focalizzato all'adattamento dei parametri di configurazione ed alla modifica di alcune logiche che prendere in considerazione le limitazioni introdotte dalla piattaforma di simulazione iTETRIS rispetto ad quella NS-3 potrebbe portare ad aumentarne le prestazioni ed a ridurre le differenze riscontrate fra le due implementazioni.

È oltretutto estremamente importante comprendere che in nessun caso è possibile raggiungere una configurazione ottimale valida per qualsiasi tipologia di scenario ed allo stesso tempo indipendente dalle contingenti caratteristiche di traffico, mobilità e composizione della popolazione dei nodi. Si potrebbe quindi ricercare un insieme di diversi profili di configurazione validi ciascuno per uno specifico caso d'uso. Più in generale, si potrebbe prevedere la realizzazione di un componente che offra funzionalità di configurazione dinamica per ogni elemento del sistema. Tale modulo dovrebbe cioè provvedere ad una costante messa a punto automatica di alcuni attributi sfruttando informazioni di contesto sul sistema circostante o sullo stato attuale del nodo stesso.

Strategie proattive per la prevenzione di collisioni tra pacchetti

Nel corso del capitolo 5, nel commentare i vari risultati prodotti dal sistema, si è visto come il maggiore rischio per la comunicazione dei vari nodi presenti nell'ambiente sia dato in buona parte dalla possibilità che le trasmissioni dei messaggi inviati hanno di collidere tra loro in mancanza di un adeguato controllo degli accessi al canale fisico. Nei protocolli realizzati si è sempre tenuto conto della probabilità non trascurabile che ha un messaggio di non venire recapitato e di conseguenza le logiche cercano di tollerare per quanto possibile la mancata ricezione di alcuni pacchetti.

Queste considerazioni sono valide soprattutto per il protocollo decentralizzato, infatti anche la semplice politica di controllo del canale fisico attuata dalla RSU in quello centralizzato ha permesso di ridurre significativamente la probabilità di collisione delle trasmissioni, come verificato nei risultati. Una funzionalità che contribuirebbe ad incrementare ulteriormente le performance e l'affidabilità dell'architettura potrebbe essere una maggiore context awareness da parte dei singoli nodi in quei momenti dove è richiesto un frequente invio di messaggi da parte di più nodi in un breve lasso di tempo, eventualità che si presenta ad esempio durante l'avvicinamento a un incrocio controllato da una RSU.

Come semplice dimostrazione di tale concetto si consideri la procedura di esplorazione dei nodi nel protocollo decentralizzato descritta al paragrafo 4.1.2.1. Tale fase è estremamente importante per gestire correttamente il ciclo di vita di un plotone e

permette allo stesso tempo di recuperare i dati di interesse dai vari nodi membri. È già stato sottolineato come tale procedura comporti l'invio di un pacchetto informativo da parte di ogni partecipante ad un gruppo entro un intervallo di tempo stabilito in maniera dinamica. La semplice modalità casuale con la quale i singoli nodi coinvolti nella procedura scelgono il momento per l'invio del proprio messaggio non è sempre sufficiente a garantire che non si verifichino numerose collisioni con pacchetti di altre trasmissioni ad opera degli stessi membri o di altri nodi nelle vicinanze. Riuscire ad organizzare in maniera più efficiente la sequenza di istanti in cui i diversi partecipanti di un gruppo devono rispondere al messaggio del loro leader porterebbe sicuramente ad una minore probabilità di collisione, ridurrebbe il tempo necessario per ogni probe e nel complesso renderebbe l'intero protocollo più stabile e performante.

Una semplice modifica nel protocollo centralizzato volta a diminuire la probabilità di collisioni è stata introdotta in calce al capitolo 4.2.1: il punto critico in questa logica è costituito dalla scelta casuale dell'istante di risposta al beacon attuata da un nodo per comunicare il proprio stato. Come illustrato l'estensione di questo intervallo è stabilita dalla RSU che potrebbe dimensionarlo dinamicamente per tenere conto sia della topologia stradale che dell'attuale stato del traffico con lo scopo di concedere più tempo nelle situazioni di grande concentrazione di veicoli.

Più in generale, per incrementare ulteriormente la qualità complessiva del sistema, andrebbe promosso ogni genere di sforzo da parte di un qualsiasi nodo per limitare o regolamentare l'invio dei propri messaggi, sfruttando a tale scopo informazioni riguardanti la propria posizione, la presenza di altri veicoli nelle vicinanze o comunque qualsiasi tipo di conoscenza del contesto nel quale esso si trova ad operare.

Estensioni ai protocolli volte a migliorarne le capacità

Ci sono molte possibilità di estensione dei protocolli presentati volte ad aggiungere funzionalità o ad ottimizzarne il funzionamento. Di seguito ne saranno illustrate alcune che per limiti di tempo non sono state implementate durante il lavoro svolto in questo progetto.

Nel protocollo decentralizzato è possibile implementare una comunicazione fra i raggruppamenti di veicoli: per come è stato definito il concetto di gruppo nell'architettura presentata, esso risulta un'entità autonoma ed isolata da altri eventuali plotoni circostanti, inclusi quelli appartenenti alla stessa direzione di riferimento. Potrebbero in realtà verificarsi numerose situazioni nelle quali una forma di comunicazione tra i differenti gruppi può contribuire a migliorare considerevolmente la resa complessiva dei protocolli sviluppati.

L'adozione di tecniche che realizzino una collaborazione tra gruppi in grado di comunicare direttamente o indirettamente offre la possibilità di migliorare notevolmente la qualità dei dati prodotti dal sistema e di sintetizzarne di nuovi. Si otterrebbe infatti la capacità di poter sfruttare un secondo livello di data fusion integrando le informazioni non soltanto tra i veicoli raggruppati ma anche tra gruppi di vetture presenti in un'area anche relativamente vasta.

Un'altra motivazione per implementare questa collaborazione potrebbe essere quella di migliorare la coordinazione delle comunicazioni. Come introdotto precedentemente, la mancanza di una politica di controllo del canale fisico aumenta le collisioni fra le trasmissioni e di conseguenza riduce le prestazioni ottenute. Una coordinazione fra i gruppi volta a diminuire questo problema potrebbe coordinarne le operazioni per evitare, ad esempio, situazioni nelle quali due o più leader sovrappongano una procedura di esplorazione dei loro membri causando un picco di comunicazioni che porta ad un inevitabile altra probabilità che si verifichino collisioni.

I messaggi di beacon inviati dalle RSU potrebbero essere estesi per prevedere la possibilità di comunicare ai nodi informazioni di contesto, oltre a quelle necessarie per il protocollo di raccolta dei dati. Potrebbe essere inserita, ad esempio, una indicazione sulla velocità consigliata ai veicoli per attraversare l'incrocio. Questa potrebbe derivare da una politica di sincronizzazione fra semafori consecutivi, la cosiddetta onda verde, che permette di ridurre il tempo trascorso in coda dai veicoli, con il conseguente miglioramento del flusso del traffico e riduzione dei consumi [Kel12].

Il protocollo centralizzato può essere modificato per eliminare uno dei limiti evidenziati durante la fase di raccolta dei risultati, costituito dall'assenza di una politica di routing dei messaggi che riduce la sua area di funzionamento, soprattutto nel caso di utilizzo di dispositivi trasmissivi con funzionalità limitate. Nella creazione di questa funzionalità occorre prestare particolare attenzione a non aumentare eccessivamente la complessità del protocollo introducendo una politica complessa e difficilmente giustificabile per la raccolta di informazioni sul traffico.

Un semplice protocollo di routing geografico potrebbe essere il seguente: un nodo che riceve una risposta ad un beacon inviata da un altro veicolo può rispondere a sua volta solamente nel caso in cui non abbia ancora ricevuto messaggi da quella particolare RSU e se la sua posizione è più lontana dall'incrocio rispetto a quella del mittente del pacchetto ricevuto. Nella nuova risposta il nodo inserisce l'id del veicolo che lo ha attivato, che si farà carico di consegnare il messaggio alla RSU. Per limitare il traffico di rete si può prevedere un consegna di ritardata: quando un nodo riceve una

risposta da un veicolo da lui attivato lo memorizza in attesa di consegnarlo alla RSU nella sua prossima risposta ad un beacon. Questa semplice politica di routing permetterebbe di ampliare il raggio di copertura del protocollo centralizzato senza aumentare eccessivamente il numero di messaggi scambiati e garantendo comunque il controllo del canale trasmissivo alla RSU.

Entrambi i protocolli potrebbero inoltre essere estesi per fornire informazioni sui tratti stradali in uscita dalla RSU. In questo caso il protocollo decentralizzato sarebbe meno adatto alla raccolta di questi dati in quanto occorrerebbe ricreare i raggruppamenti di veicoli, aumentando l'utilizzo del canale fisico di comunicazione ed introducendo un ritardo che potrebbe ridurre la quantità raccolta. Il protocollo centralizzato si può adattare meglio per assolvere tale compito: è sufficiente modificare il formato dei messaggi di beacon per indicare se si è interessati alle informazioni dei nodi che si avvicinano o allontanano dall'incrocio. In questo modo è possibile creare messaggi beacon differenti per una stessa direzione che richiedono l'attivazione dei soli nodi sono in avvicinamento o in allontanamento dalla RSU.

Estensione dell'applicazione iTETRIS realizzata

Il vantaggio principale della piattaforma iTETRIS utilizzata in questo progetto rispetto ad altri ambienti, è l'integrazione di un simulatore di comunicazioni wireless ed uno di mobilità per testare protocolli di VANET nel modo più corretto ed accurato possibile. Il sistema presentato in questo progetto non sfrutta tutte le funzionalità messe a disposizione da questa architettura: è prevista la sola raccolta di informazioni sul traffico senza nessun tipo di controllo sulla mobilità dei veicoli che seguono la traccia specificata nel particolare scenario di simulazione utilizzato.

Una estensione di questo progetto può quindi essere volta alla creazione di protocolli evoluti ed intelligenti per l'elaborazione delle informazioni raccolte con lo scopo di realizzare un controllo delle logiche semaforiche dinamico basato sul reale stato del traffico. Questi nuovi sistemi possono poi essere testati per verificarne l'efficacia sulla piattaforma iTETRIS che permette la modifica della viabilità dei stradale interagendo direttamente sui vari parametri dello scenario utilizzato.

L'interfaccia per iCS realizzata in questo progetto può servire inoltre come base per lo sviluppo di nuovi protocolli di comunicazione veicolare, che possono essere sviluppati più semplicemente senza la necessità di avere intima familiarità con il funzionamento interno di questa piattaforma di simulazione.

Conclusioni

Il lavoro di tesi presentato in questo testo ha approfondito ed esteso le conoscenze acquisite nel corso di Sistemi Mobili M e si è proposto di metterle in pratica mediante l'analisi del problema reale discusso nel capitolo 1 e la successiva sperimentazione di un insieme di tecniche e protocolli in grado di rispondere adeguatamente alle esigenze sollevate.

Il progetto intrapreso ha quindi permesso di acquisire una preziosa esperienza con alcuni importanti strumenti di simulazione open-source estremamente diffusi in ambito accademico. L'intera architettura software proposta e presentata nel capitolo 2 è stata completamente implementata e collaudata all'interno dell'ambiente di simulazione di comunicazione veicolare offerto dal pacchetto software iTETRIS, con le modalità esposte nel corso dei capitoli 3 e 4. La successiva fase di collaudo ha richiesto l'assimilazione di alcune nozioni di base sull'utilizzo dei simulatori di rete NS-3 e di mobilità SUMO che ne compongono l'architettura al fine di ottimizzare le prestazioni ottenute negli scenari urbani utilizzati.

Il lavoro di tesi presentato si è articolato principalmente su tre fasi. Una prima parte è stata dedicata allo studio del funzionamento della piattaforma di simulazione utilizzata al fine di creare un modulo di interfaccia per facilitare lo sviluppo di protocolli di comunicazione veicolare, cercando di realizzare una separazione netta fra le modalità di interazione che caratterizzano il simulatore e le particolari logiche applicative realizzate, dedicate allo scambio dei dati ed alla loro elaborazione.

Questo permette la creazione e la verifica di nuove soluzioni nel campo di Intelligent Transportation System (ITS) più semplicemente e velocemente su una piattaforma specificamente pensata per simulare in modo realistico sistemi di comunicazione veicolare.

In seguito si è proceduto alla realizzazione di una implementazione di un protocollo decentralizzato per il raggruppamento dei veicoli che utilizzasse il modulo di interfaccia al simulatore sviluppato, usando come base il lavoro svolto in precedenza da altri studenti di questo corso. Durante la realizzazione e la successiva verifica di

questo sistema si sono incontrate numerose limitazioni nella piattaforma di simulazione utilizzata che hanno richiesto una modifica mirata alla risoluzione o alla limitazione degli effetti causati dai problemi emersi.

Utilizzando all'esperienza acquisita dal lavoro svolto si è poi passati alla progettazione ed alla realizzazione un protocollo di raccolta dati completamente nuovo, che traesse beneficio dai moduli precedentemente sviluppati, andando a migliorare i punti deboli evidenziati durante la sperimentazione del protocollo decentralizzato. Questo ha comportato la creazione di una soluzione originale che permettesse di ottenere ottimi risultati nonostante la semplicità delle logiche utilizzate.

I numerosi test condotti ed in particolare quelli esposti nel capitolo 5, dimostrano come, nonostante il raggruppamento dei veicoli realizzato nel protocollo decentralizzato sia in grado portare alla formazione di plotoni di veicoli di ottima qualità, per una raccolta ottimale di dati sulla mobilità dei nodi nelle vicinanze di un punto di interesse sia preferibile l'utilizzo di protocolli più semplici, nei quali l'aggregazione fra i veicoli viene eliminata per favorire la coordinazione delle loro comunicazioni, le quali incidono in maniera diretta sulla qualità dei dati raccolti.

Per contro l'utilizzo di raggruppamenti di veicoli può essere di grande importanza per realizzare le funzionalità avanzate previste nei sistemi di Intelligent Transportation come la sicurezza attiva fra veicoli o la disseminazione di informazioni.

A causa della natura prettamente sperimentale del progetto, l'architettura appena esposta presenta alcuni punti di forza ma concede abbondante spazio ad ulteriori sviluppi, correzioni e approfondimenti possibili. Nel corso del capitolo 5 si sono evidenziate alcune imperfezioni insite nel protocollo decentralizzato e sono state fornite indicazioni su come sia possibile migliorarlo per poterlo rendere maggiormente competitivo in termini di numero di messaggi scambiati. Si sono inoltre evidenziate le limitazioni derivanti dalla scelta di semplificare la modalità di comunicazione del protocollo centralizzato ed è stato proposto un possibile miglioramento volto ad estendere l'area di funzionamento.

In calce al capitolo 6 sono state introdotte numerose possibilità di espansione ed ottimizzazione per il sistema complessivo, molte delle quali sono frutto dell'esperienza maturata nel corso della sua implementazione e messa a punto. In particolare viene suggerita una soluzione volta all'estensione dei protocolli per l'integrazione con applicazioni di ITS per la gestione del flusso del traffico.

Appendice A

Parametri di configurazione del sistema

In questa appendice sono indicati tutti parametri di configurazione per i ogni componente del sistema ed indicati i valori di default utilizzati durante le simulazioni.

Interfaccia iCS

Tabella 1: Attributi della classe `Node`.

Attributo	Tipo	Default	Descrizione
P.Full	double	1	Probabilità per la classe full di veicoli.
P.Medium	double	0	Probabilità per la classe media di veicoli.
RadiusRSU	double	170m	Raggio di propagazione delle RSU.
RadiusFull	double	170m	Raggio di propagazione della classe full.
RadiusMedium	double	170m	Raggio di propagazione della classe media.

Controllore

Tabella 2: Attributi della classe `ItsController`.

Attributo	Tipo	Default	Descrizione
Type	enum	-	Tipologia del nodo che si intende rappresentare (RSU, veicolo full o medium).
Direction-Tolerance	double	8.0°	Angolo di tolleranza per il cono di conformità per la direzione di questo nodo. Non utilizzato per le RSU.

Node Sampler

Tabella 3: Attributi della classe `NodeSampler` comuni ad ogni tipo.

Attributo	Tipo	Default	Descrizione
Quantity	intero	5	Numero di campioni memorizzati all'interno del buffer circolare.
Movement-Threshold	double	2.0m	Distanza minima tra due campionamenti successivi.
SpeedVariance	double	0.25	Valore della varianza per la distribuzione degli errori del modulo della velocità.

Tabella 4: Attributi della classe `NodeSampler` utilizzati per i veicoli full.

Attributo	Tipo	Default	Descrizione
Resolution	intero	1000ms	Intervallo di tempo tra due campionamenti successivi.
Position-Radius	double	20.0m	Raggio di errore massimo per il campionamento della posizione.
Position-Variance	double	1.8	Valore della varianza per la distribuzione degli errori di posizione.
Direction-Variance	double	0.002	Valore della varianza per la distribuzione degli errori di direzione.
SpeedError	double	0.15m/s	Valore medio di errore sulla velocità [Wit05].

Tabella 5: Attributi della classe `NodeSampler` utilizzati per i veicoli medium.

Attributo	Tipo	Default	Descrizione
Resolution	intero	1000ms	Intervallo di tempo tra due campionamenti successivi.
Position-Radius	double	20.0m	Raggio di errore massimo per il campionamento della posizione.
Position-Variance	double	7.2	Valore della varianza per la distribuzione degli errori di posizione.
Direction-Variance	double	0.008	Valore della varianza per la distribuzione degli errori di direzione.
SpeedError	double	1.5m/s	Valore medio di errore sulla velocità [Han14].

Protocollo decentralizzato

Tabella 6: Attributi della classe BehaviourGroupSetupReactive.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
TimeResponseMax	intero	15ms	Intervallo massimo di attesa prima dell'invio di messaggi di risposta o di offerta (nel caso di controfferta).
TimeBidMin	intero	150ms	Intervallo minimo di attesa prima del passaggio dalla prima alla seconda fase.
TimeBidMax	intero	250ms	Intervallo massimo di attesa prima del passaggio dalla prima alla seconda fase.
TimeLeaderMin	intero	100ms	Intervallo minimo di attesa prima dell'invio del messaggio di leadership.
TimeLeaderMax	intero	200ms	Intervallo massimo di attesa prima dell'invio del messaggio di leadership.
MaxHops	intero	2	Numero massimo di salti di attivazione consentiti.

Tabella 7: Attributi della classe BehaviourGroupSetupProactive.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
DelaySkew	intero	20ms	Valore massimo che viene aggiunto o sottratto per differenziare agli intervalli temporali scelti casualmente.
DelayBase	intero	200	Intervallo base di attesa per il calcolo del tempo di attesa in caso di invito.
WaitExponent	double	0.5	Esponente per il calcolo del tempo di attesa in caso di invito.
Timeout	intero	250ms	Periodo di timeout successivo all'invio del messaggio di invito.

Tabella 8: Attributi della classe `ItsGroupController`.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
GroupSetupMode	intero	-	Strategia di formazione dei gruppi: - 0: strategia reattiva. - 1: strategia proattiva.
SinkThreshold	double	20m	Soglia superata la quale il leader del gruppo ne comanda la terminazione.

Tabella 9: Attributi della classe `BehaviourRSU`.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
TimeBeacon	intero	500ms	Tempo fra due messaggi di beacon successivi.
TimeCheck	intero	1000ms	Intervallo fra le esecuzioni dell'euristica di manipolazione dei dati ricevuti.
Timeout	intero	3000ms	Tempo concesso prima di eliminare un gruppo a causa della mancata risposta ai più messaggi beacon consecutivi.

Tabella 10: Attributi della classe `BehaviourNodeMap`.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
RsuSilence	booleano	true	Impedisce l'invio di messaggi heartbeat se viene rilevata la presenza di una RSU.
Timeout	intero	4100ms	Timeout massimo superato il quale un nodo viene rimosso dalla collezione se non sono ricevute informazioni sul suo stato.
TimeCheck	intero	500ms	Intervallo di tempo che separa due controlli successivi della nodemap.
TimeHeartbeat	intero	2000ms	Intervallo di tempo che separa l'invio dei messaggi di heartbeat.

Tabella 11: Attributi della classe BehaviourGroupJoin.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
JoinDelayMin	intero	15ms	Intervallo minimo di attesa prima di unirsi ad un gruppo compatibile.
JoinDelayMax	intero	80ms	Intervallo massimo di attesa prima di unirsi ad un gruppo compatibile.

Tabella 12: Attributi della classe BehaviourGroupLeader.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
MergeGroup	booleano	true	Stabilisce se al leader è permesso unirsi (in qualità di membro) a un altro gruppo se il proprio risulta vuoto.
TimeProbeMin	intero	150ms	Periodo minimo per la durata della procedura di esplorazione dei membri.
TimeProbeMax	intero	1000ms	Periodo massimo per la durata della procedura di esplorazione dei membri.
TimeProbeSkew	intero	50ms	Valore massimo che viene aggiunto o sottratto al tempo di attesa tra due esplorazioni consecutive.
MaxProbeFails	intero	1	Massimo numero di iterazioni di esplorazione che un nodo membro può fallire prima di venire espulso dal gruppo.

Tabella 13: Attributi della classe BehaviourGroupMember.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
QuickDispose	booleano	true	Stabilisce se al nodo è permesso abbandonare il gruppo se rileva che la distanza dalla RSU di riferimento è in aumento.
LeaderTimeout	intero	2000ms	Periodo di timeout per il leader del gruppo.

Protocollo centralizzato

Tabella 14: Attributi della classe BehaviourRSU.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
TimeBeacon	intero	1000ms	Durata di un ciclo di esplorazione delle differenti direzioni attraverso beacon successivi.
TimeBeacon-Min	intero	250ms	Tempo minimo fra due beacon successivi.
TimeCheck	intero	1000ms	Intervallo fra le esecuzioni dell'euristica di manipolazione dei dati ricevuti.
Timeout	intero	3000ms	Tempo concesso prima di eliminare un nodo a causa della mancata risposta ai più messaggi beacon consecutivi.

Tabella 15: Attributi della classe BehaviourNode.

Attributo	Tipo	Default	Descrizione
Enabled	booleano	true	Attiva il componente durante la simulazione.
ResponseDelay	intero	10ms	Delay utilizzato per evitare la sovrapposizione fra messaggi di beacon e le risposte dei nodi.
SinkThreshold	double	20m	Soglia superata la quale il nodo sospende l'invio delle risposte.
AvgSample	intero	15	Numero di campionamenti utilizzati per la media della velocità del nodo.

Bibliografia

- [ASP08] Azarmi, M., Sabaei, M., Pedram, H. «Adaptive routing protocols for vehicular ad hoc networks». In: Telecommunications, 2008. IST. International Symposium on. pp. 825–830. doi: 10.1109/ISTEL.2008.4651414.
- [Ble12] Bley, O. et al. «Improvement in traffic state estimation at signal controlled intersections by merging induction loop data with V2X data». In: Archives of Transport System Telematics Vol. 5, iss.3 (2012), pp. 3–7.
- [Coh14] COLOMBO Project homepage. 2014. url: <http://www.colombo-fp7.eu/>.
- [CSZ11] Cheng, H. T., Shan H., Zhuang, W. «Infotainment and road safety service support in vehicular networking: From a communication perspective». In: Mechanical Systems and Signal Processing 25.6 (2011), pp. 2020–2038. issn: 0888-3270. doi: 10.1016/j.ymssp.2010.11.009.
- [Dan03] Dana, P. H. «Geodetic Datum Overview». 2003. url: http://www.colorado.edu/geography/gcraft/notes/datum/datum_f.html.
- [ETS13] ETSI EN 302 637-2. «Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications». A cura di Martin Arndt. Version 1.3.0. Sophia Antipolis, France: ETSI ITS WG2, 2013.
- [FLK11] Faouzi, N. E., Leung, H., Kurian, A. «Data fusion in intelligent transportation systems: Progress and challenges – A survey». In: Information Fusion 12.1 (2011). Special Issue on Intelligent Transportation Systems, pp. 4–10. issn: 1566-2535. doi: 10.1016/j.inffus.2010.06.001.
- [GSL07] Guberinic, S., Senborn, G., Lazic, B. «Optimal Traffic Control: Urban Intersections». Taylor & Francis, 2007. isbn: 9781420062700.
- [Han14] Han, H., et al. «SenSpeed: Sensing driving conditions to estimate vehicle speed in urban environments» INFOCOM, 2014 Proceedings IEEE, May 2014.
- [HL01] Hall, D. L., Llinas, J. «Handbook of Multisensor Data Fusion». CRC Press, 202001. 2008. isbn: 0849323797.
- [HXG02] Hong, X., Xu, K., Gerla, M. «Scalable routing protocols for mobile ad hoc networks». In: Network, IEEE 16.4 (2002), pp. 11–21. issn: 0890-8044. doi: 10.1109/MNET.2002.1020231.

- [Itb14] iTETRIS Building, Installation and Configuration Guidelines. url: <http://www.ict-itetris.eu/10-10-10-community/>.
- [Ith14] iTetris homepage. 2014. url: <http://www.ict-itetris.eu/>.
- [Kel12] Kelly, B. «A ‘Green Wave’ Reprieve». In: Traffic Engineering & Control Volume: 53 iss: 2 2012 issn: 0041-0683.
- [Kra12] Krajzewicz, D. et al. «Recent Development and Applications of SUMO - Simulation of Urban MObility». International Journal On Advances in Systems and Measurements, 5 (3&4):128-138, 2012.
- [Kra13] Krajzewicz, D. et al. «COLOMBO: Investigating the Potential of V2X for traffic management purposes assuming low penetration rates». In: 9th ITS European Congress. 2013.
- [KTW08] Krajzewicz, D., Teta Boyom, D., Wagner, P. «Untersuchungen der Performanz einer auf C2C-Kommunikation basierenden, autonomen Routenwahl bei Stauszenarien», In: Heureka 2008. Stuttgart (Germany).
- [Jur12] Jurgen R. K. «V2V/V2I Communications for Improved Road Safety and Efficiency». Warrendale, PA: SAE International, 2012. isbn: 978-0-7680-7725-4. doi: 10.4271/pt-154.
- [Man09] Maneros J. et al. «iTETRIS Platform Architecture for the Integration of Cooperative Traffic and Wireless Simulations». ITS-T 2009, Lille (France).
- [NSh14] NS-3 homepage. 2014. url: <http://www.nsnam.org/>.
- [NSt14] NS-3 Tracing Manual. 2014. url: <http://www.nsnam.org/docs/manual/html/tracing.html>.
- [PG11] Perahia, E., Gong, M. X. «Gigabit Wireless LANs: An Overview of IEEE 802.11Ac and 802.11Ad». In: SIGMOBILE Mob. Comput. Commun. Rev. 15.3 (2011), pp. 23–33. issn: 1559-1662. doi: 10.1145/2073290.2073294.
- [Ran94] Rankin, J. «An error model for sensor simulation GPS and differential GPS». In: Position Location and Navigation Symposium, 1994. IEEE. pp. 260–266. doi: 10.1109/PLANS.1994.303322.
- [Ron13] Rondinone M. et al. «iTETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications» 2013 doi: 10.1016/j.simpat.2013.01.007.

- [Sad11] Sadrozinski, H. «Applications of field-programmable gate arrays in scientific research». Boca Raton, FL: CRC Press, 2011.
- [SK10] Sarik, J., Kymissis, I. «Lab kits using the Arduino prototyping platform». In: Frontiers in Education Conference, 2010. IEEE. doi: 10.1109/FIE.2010.5673417.
- [Tan02] Tanenbaum, A. «Computer Networks». 4th. Prentice Hall, 2002. isbn: 0130661023.
- [Wit05] Witte, T.H., et al. «Accuracy of WAAS-enabled GPS for the determination of position and speed over ground» Journal of Biomechanics, Volume 38, Issue 8, pp. 1717-1722, 2005.